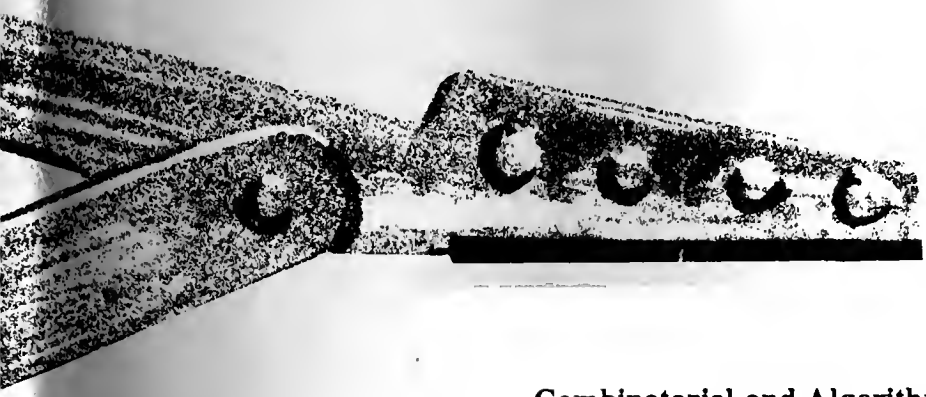
Robotics Research Technical Report

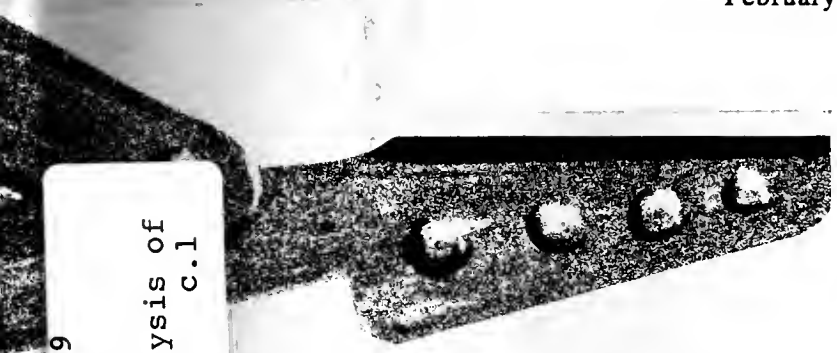


Combinatorial and Algorithmic Analysis of Space Decomposition Problems

by

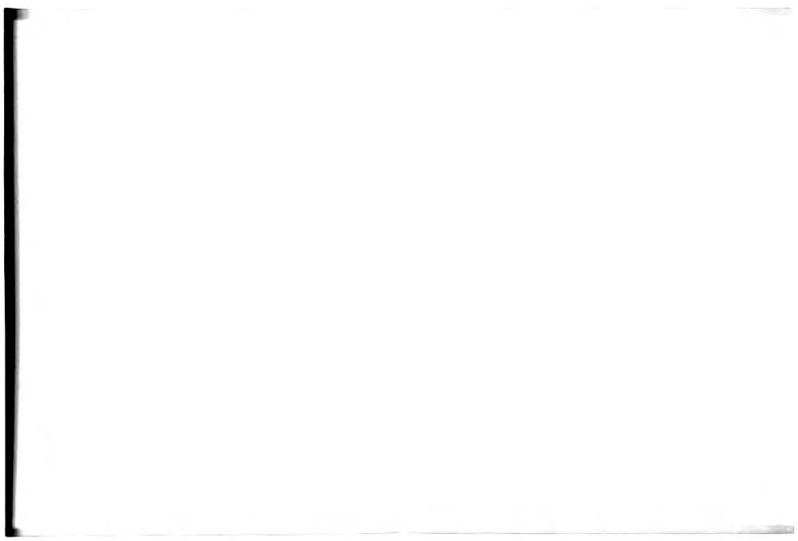
Boris Aronov

Technical Report No. 429
Robotics Report No. 189
February, 1989



New York University
ant Institute of Mathematical Sciences

Computer Science Division
251 Mercer Street New York, N.Y. 10012



Combinatorial and Algorithmic Analysis of Space Decomposition Problems

by

Boris Aronov

Technical Report No. 429
Robotics Report No. 189
February, 1989

New York University
Dept. of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, New York 10012

Work on this paper has been supported by Office of Naval Research Grant N00014-87-K-0129 National Science Foundation CER Grant DCR-83-20085, National Science Foundation Grant subcontract CMU-406349-55586, and by grants from the Digital Equipment Corporation and the IBM Corporation.

Combinatorial and algorithmic analysis of space decomposition problems

Boris Aronov

February 1989

A dissertation in the Department of Computer Science submitted to the faculty of the Graduate School of Arts and Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy at New York University.

Approved

(Signed) _____
Micha Sharir,
Research Advisor

Combinatorial and algorithmic analysis of space decomposition problems

Boris Aronov

Abstract

The first part of the thesis studies geodesic Voronoi diagrams. The closest-site (respectively, furthest-site) Voronoi diagram of a finite set of *sites* in Euclidean space is a classical geometric structure, which partitions the space into a set of *Voronoi cells*, each associated with a site, so that any point in the cell of site s is closer to s (resp. further from s) than to any other site. The structure of such diagrams for point sites in the plane has been completely characterized and well-known efficient algorithms exist for computing them.

Voronoi diagrams have been generalized by replacing the Euclidean distance by a more general metric and/or relaxing the assumption that sites be single points. We consider the closest- and the furthest-site Voronoi diagrams for a set of k point sites in a simple n-gon, defined by the internal geodesic distance inside the polygon. We demonstrate that the planar map defined by either diagram is comprised of $O(n + k)$ features of bounded complexity each and describe nearly optimal algorithms for constructing the two Voronoi diagrams. Namely, the closest-site geodesic Voronoi diagram can be computed in time $O((n+k)\log(n+k)\log n)$, while $O((n+k)\log(n+k))$ time is sufficient for the furthest-site diagram.

The second part of the thesis analyzes the structure of an arrangement of flat triangles in 3-space. The combined combinatorial complexity of all non-convex cells (i.e., non-convex components of the complement of the union of the triangles), maximized over all arrangements of n triangles is shown to be roughly $O(n^{\frac{7}{3}})$, improving the best previously known upper bound of $O(n^{3-\frac{1}{49}})$ for a smaller quantity—the maximum combinatorial complexity of a *single cell.*

Our result has applications to algorithmic motion planning, stemming from the well-known technique that transforms a polyhedral body translating in a polyhedral environment into a collection of convex polygonal plates in three-dimensional space; the set of placements of the body reachable from a starting configuration along a collision-free path corresponds to a cell in the arrangement of these plates. Thus analyzing the maximum combinatorial complexity of a single cell and obtaining a comparably efficient algorithm for its calculation constitutes a satisfactory solution to the translational motion planning just mentioned.

To this end, we also consider the problem of computing a single cell or a subset of cells in a three-dimensional arrangement of triangles, providing a nearly worst-case optimal randomized algorithm for solving the former problem and a less efficient procedure for the latter. In addition, we examine a few special classes of arrangements for which better estimates on the maximum single-cell complexity can be deduced and where computing a cell or any collection of cells appears easier.

Author's Current Address

Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street, New York, NY 10012 USA
E-mail Address: aronov@acf4.nyu.edu

Acknowledgments

I could not possibly list all individuals who have contributed to my completion of this thesis. At the risk of omission, let me mention a few to whom I feel most indebted.

First and foremost, I wish to thank my thesis advisor, Micha Sharir, for his infinite patience and untiring guidance throughout these years. He taught me much about research.

I also wish to express my gratitude to Steven Fortune and Gordon Wilfong for their support and encouragement. The generosity expressed by their capacity to listen patiently and thoughtfully to my ideas, many of which were half-baked, surpassed most reasonable expectations. It was they who made my two summers at AT&T Bell Laboratories a valuable learning experience which grew into a continuing productive collaboration.

I am grateful to Colm Ó'Dúnlaing who introduced me to the subject of Computational Geometry and guided me through my first research efforts. I am greatly indebted to Richard Cole for his valuable advice above and beyond the call of duty. I am also indebted to Richard Pollack for the numerous valuable discussions and for his guidance and support.

The "look" of this thesis has been much influenced by Pankaj K. Agarwal, Leo Joskowicz, and Ofer Zajicek, while Steven Fortune contributed a great deal to its "feel." Support and encouragement of these and other friends and colleagues have helped me a great deal through these years.

Last, but not least, I would not be where and who I am today, were it not for my parents, Aron and Yulia Aronov, and my sister Irina. I am most grateful for their help and encouragement.

The research for this thesis was partially supported by an AT&T Bell Laboratories Ph.D. Scholarship. Part of the work was performed during the summer of 1987, which I spent at AT&T Bell Laboratories in Murray Hill, New Jersey. The analysis

of relative convex hulls and furthest-site geodesic Voronoi diagrams constitutes joint work with Steven Fortune and Gordon Wilfong of Bell Laboratories, while the second part of the thesis is a collaboration with Micha Sharir.

Contents

List of Figures

Chapter 1

Introduction

> The Red Queen shook her head.
> "You may call it "nonsense" if you like," she said,
> "but *I've* heard nonsense, compared with which
> that would be as sensible as a dictionary!"
>
> Lewis Carroll, *Through the Looking Glass*

Computational geometry is the study of algorithm design for geometric problems. It combines computer science techniques of efficient data structure design with combinatorial analysis of the underlying geometric structure. As in the case of algorithms for purely combinatorial problems, these two themes interact and provide a rich mathematical structure. This thesis is concerned with "space decomposition problems," which is a general concept that encompasses a broad spectrum of geometric problems. In this thesis we present the combinatorial analysis of several space decomposition problems and apply it to devise efficient algorithms for the problems under consideration.

The concept of a "space decomposition problem" covers quite a few geometric as well as non-geometric questions. Probably the simplest, albeit somewhat artificial, example of such a problem is that of sorting real numbers—the required sorted order partitions the real line into segments free of input points. A much better example of a subject that, though not inherently geometric, is often viewed as a space partitioning problem, is that of object classification. Namely, in a family of objects

under consideration, one identifies a finite collection of quantitative parameters, each presumably measuring a different aspect of an object, thus establishing a map from objects to be classified to points in the resulting *feature space*. Now the problem of classification of a new object based on previously seen representatives of various classes translates into constructing a partition of the feature space consistent with the classification of given representative points (for an overview of the subject, see, for example [Eve74]). The question of the choice of the feature space partitioning criterion is entirely outside the scope of our present discussion; however, once it is made, the following algorithmic question arises—one would like to compute this space decomposition in an efficient manner that also facilitates subsequent queries of the form "To which class does the query object belong?" (The latter question is an instance of the all too familiar point-location problem.) Finally, let us mention the *post-office problem*[Knu73], which is closely related to one of the questions considered in this thesis: "Given a set of post-offices and one's current location, identify the nearest post-office." The set of post-offices induces a natural partition of the plane into "postal zones", each containing points closest to a particular post office—this is precisely the Voronoi diagram, to be discussed in considerable detail below. The answer to the query (again, a point location query) is then obtained by identifying the "postal zone" containing one's location.

This thesis discusses two quite different variations on the space decomposition theme. One is concerned with the classical geometric notion of a Voronoi diagram in the somewhat non-standard context of the internal geodesic metric in a simple polygon, while the other deals with the decomposition of three-dimensional Euclidean space induced by a collection of intersecting flat triangles. The latter problem arises in translational motion planning with three degrees of freedom—the connection is sketched below and discussed in more detail in Chapter 5.

Given a finite set of point *sites* in the plane or, more generally, in Euclidean space of any dimension, a classical geometric structure associated with them is that of the *Voronoi diagram* (also known as the Thiessen tesselation or Dirichlet complex). It

can be viewed as the decomposition of the space into (open) *Voronoi cells*, one cell per site, so that the cell of site s consists of points that are closer to s than to any other site. The Voronoi diagram proper is defined as the complement of the (open) Voronoi cells, i.e., the collection of all points x such that the set of sites closest to x consists of two or more sites. Alternatively, the Voronoi structure is sometimes identified with the partition of the space into maximal sets with the property that the collection of sites closest to a point in such a set is independent of the choice of the point.

The structure of Voronoi diagrams has been extensively studied in the past (for a comprehensive list of references, see [Ede87,Aur88]). In fact, Voronoi cells form a decomposition of space into convex polytopes; the cell of site s is the intersection of the half spaces $H(s,t)$, for all sites $t \neq s$, where $H(s,t)$ is the locus of points lying closer to s than t. The Voronoi diagram proper, being the complement of the union of Voronoi cells, is easily seen to be a d-complex in $(d+1)$-space. Both the geometric structure and the combinatorial complexity (which in this case refers to the total number of faces of all dimensions) of a Voronoi diagram have been completely characterized (see, for example, [Ede87]). In fact, the geometric structure of the diagram in $\boldsymbol{R}^d$ is isomorphic to the structure of the (lower) convex hull of a set of points in $\boldsymbol{R}^{d+1}$ lying in convex position, and thus its worst-case combinatorial complexity is $\Theta(n^{\lceil d/2 \rceil})$, where n is the number of sites (for a more detailed treatment of this subject, see [Ede87, Chap. 13]). In particular, the issue of computing the Voronoi diagram of n points in $\boldsymbol{R}^d$ reduces to that of calculating the convex hull of n points in $\boldsymbol{R}^{d+1}$ which, via duality, is equivalent to determining the lower envelope of n hyperplanes in $\boldsymbol{R}^{d+1}$. This problem can be solved in time $O(n \log n)$ for $d \leq 2$ and time $O(n^{\lceil (d+1)/2 \rceil})$ for $d \geq 3$, which is optimal for $d = 2$ and all odd d [Ede87, Chap. 13]. The question of computing the Voronoi diagram of point sites *in the plane* has been extensively studied, producing a large variety of worst-case optimal algorithms (for instance, [SH75,Bro79,For87]).

It is interesting to observe how the structure and/or complexity of the Voronoi

diagram changes when one replaces its classical definition described above by some less standard generalization. There are two natural directions for possible generalization—one could vary the space being partitioned (in terms of modifying its metric and/or underlying set of points) or consider a different class of *sites* to replace points.

While the structure and complexity of classical Voronoi diagrams in Euclidean spaces of arbitrary dimension are well-understood, either of the above generalizations results in a notion of Voronoi diagram whose properties even in the simple two-dimensional case may be quite different. Replacing the Euclidean metric by an arbitrary L_p metric or, in fact, by an arbitrary convex distance function (so-called *Minkowski distance*) has been explored by [CD85,For85,LS87b], who also present efficient algorithms for computing the diagrams thus defined. Moreover, Aurenhammer [Aur88] has demonstrated that certain topological properties of Voronoi diagrams, facilitating construction of efficient algorithms for computing them, hold even if Minkowski distance is replaced by a distance function satisfying an even weaker set of assumptions. With respect to any of these distance functions, the Voronoi diagram partitions the plane into a set of cells, each of which is both connected and simply connected and contains its owner site. Since Voronoi diagram can be viewed as a planar graph, Euler's formula implies that the number of its vertices and edges (usually referred to as "Voronoi vertices" and "Voronoi edge") is linear in the number of cells, which in turn equals the number of sites. Thus the complexity of the diagram of n sites, measured as the number of Voronoi cells, edges, and vertices is linear in n. However, in the case of an arbitrary convex distance measure, a Voronoi edge, being a portion of the bisector between two sites (which is the set of points equidistant from the two sites), need not be an object of bounded complexity—for example, if the distance function is defined by a convex k-gon, the bisector is a piecewise-linear path with $\Theta(k)$ turns. Thus in this case the "fine-grain" complexity of the Voronoi diagram is bounded by $O(kn)$ [LS87b]—it is not difficult to see that this bound is asymptotically sharp in the worst case.

A Minkowski distance is in general not a metric, as there is no requirement that

it be symmetric. Nevertheless, it does satisfy the triangle inequality, which is one of the key properties that controls the appearance of the Voronoi diagram. There exist generalizations of Voronoi diagrams which replace Euclidean metric with functions that do not obey the triangle inequality. Two misleadingly similar examples of this type of a distance measure involve *weighted Euclidean distances*—namely, each site is associated with a weight, and points in the plane are classified according to which site has least *weighted distance* to them. There are two versions of this problem: in the first, weights are additive [Aur87,For87,Sha85], while in the second, multiplicative [AE84]. Notably, the former case yields a linear-size Voronoi diagram in which cells are star-shaped and edges are portions of hyperbolas, while in the latter case, the cells need not be simply connected or, for that matter, connected, edges are circular arcs, and the diagram has quadratic complexity in the worst case. The lesson seems to be that not every generalization of Voronoi diagram yields a well-behaved linear-size structure.

As for the second direction of generalization of Voronoi diagrams, the notion of a "skeleton" (also known as the "Blum's medial axis transform") of a polygon, used in graphics to describe the shape of a figure is essentially the Voronoi diagram of the segments comprising the boundary of the polygon (for further references, see [Aur88]). Kirkpatrick [Kir79] analyzed Voronoi diagrams for line-segment sites and proved that the complexity of the diagram in this case is still linear in the number of sites. Fortune [For87] and Yap [Yap87] independently developed procedures that can be used for computing such diagrams efficiently. (In fact, Yap's algorithm will also handle circular arcs.) Moreover, Leven and Sharir [LS87b] have generalized Yap's technique to work for arbitrary Minkowski distances.

The notion of Voronoi diagram discussed in the preceding paragraphs is more properly known as the *closest-site Voronoi diagram*. The complementary notion of a *furthest-site Voronoi diagram* of a set of sites defines the (open) Voronoi cell of s to consist of points from which s is the furthest among all sites. The structure of such diagrams for point sites in an arbitrary Euclidean space has been completely

characterized—it has been shown to correspond to the structure of the (upper) convex hull of a certain set of points in convex position in one higher dimension [Bro79]. For the planar case, there exists a wide variety of provably optimal algorithms for constructing the furthest-site Voronoi diagram of point sites (for example, [SH75, Bro79]).

In this thesis, in view of much recent interest in the geodesic geometry in a simple polygon (such as [GHL*87,PSR,Sur86,Sur87,Tou86]), we consider the closest- and the furthest-site geodesic Voronoi diagrams for a set of k point sites in a simple n-gon; in other words, we restrict our attention to a simple polygon and analyze the structure of the Voronoi diagram, where the distance between two points is measured by the length of the shortest path that connects them while remaining in the polygon. We demonstrate that, in a very strong sense, both diagrams have linear complexity; namely, the planar map defined by either diagram is comprised of $O(n + k)$ features of bounded complexity each. Furthermore, we describe nearly optimal algorithms for constructing the two Voronoi diagrams. The closest-site diagram can be computed in time $O((n+k)\log(n+k)\log n)$, while for the furthest-site diagram $O((n+k)\log(n+k))$ time is sufficient. In Chapter 2 we present a thorough study of the geometry of geodesic distance inside a simple polygon. Algorithms for computing the closest-site and furthest-site geodesic Voronoi diagrams are described in Chapters 3 and 4, respectively. It appears that the algorithms, though bearing some resemblance to their Euclidean-metric counterparts, are far from being an easy extension of the classical case. A possible explanation lies in the fact that the geodesic metric inside a simple polygon gives rise to a significantly richer variety of degenerate situations, as compared to the Euclidean plane. As a result, the number of special cases that have to be considered is quite substantial and arguments require a great deal of technical machinery to avoid enumerating all possible degeneracies. We would also like to point out that the two Voronoi diagram algorithms presented in Chapters 3 and 4 are the culmination of other closest/furthest problems in a simple polygon (such as the geodesic furthest-neighbors problem considered in [Sur87]). In addition, we

consider the detailed technical analysis of geodesic geometry, contained in Chapter 2 (especially the discussion of the notion of a relative convex hull in Section 2.3) one of the main contributions of our work.

In the second part of the thesis we analyze the structure of a three-dimensional arrangement of *convex plates*, which is the natural subdivision of the space induced by a set of intersecting convex two-dimensional objects, such as triangles. The objective is to estimate the combinatorial complexity of a single cell (i.e., a connected component of the complement of the union of the triangles), maximized over all arrangements of n triangles. Since the total complexity of the entire arrangement is $\Theta(n^3)$ in the worst case, the goal is to establish subcubic bounds for a single cell. The best previously known upper bound of $O(n^{3-\frac{1}{49}})$ on this quantity was obtained by a rather non-trivial application of the theory of extremal hypergraphs [PS87]. We produce a nearly asymptotically sharp bound of roughly $n^{\frac{7}{3}}$ for a larger quantity, namely the maximum total complexity of all non-convex cells in such an arrangement. Our results can be viewed as the three-dimensional version of the corresponding result in the plane, where the worst-case complexity of all non-convex faces of an arrangement of n segments has been shown to lie within a logarithmic factor of $n^{4/3}$ [EW86,AEGS].

Besides the applications mentioned below, our result is significant in itself, as it provides an affirmative solution for a special case of a more general conjecture, stating that the combinatorial complexity of a single component (cell) in an arrangement of n algebraic surfaces (or surface patches) of bounded degree in $\boldsymbol{R}^d$ is always $o(n^d)$ (by Milnor's Theorem [Mil64], the complexity of the entire arrangement is $\Theta(n^d)$ in the worst case). The general form of this conjecture, even in three dimensions, appears to be very difficult. It is not even known whether the *lower envelope* of such surfaces has subcubic complexity in general. Our results (and the weaker results of [PS87]) are the only known subcubic bounds on cell complexity (apart from the trivial bound for arrangements of planes). See Chapter 5 for more background information.

A generalization of our result for triangles to arbitrary convex plates is discussed along with some applications to algorithmic motion planning. These applications

stem from the well-known approach to motion planning in which the obstacles are expanded by the object B, whose motion is to be planned, by taking appropriate Minkowski (vector) sums of the boundary elements of the obstacles and those of B [LW79]. In three dimensions, in the case where both B and the environment are polyhedral and only translations are allowed, these Minkowski sums constitute a collection of convex plates in three-dimensional space; the set of possible configurations of the system reachable from a starting configuration along a collision-free path corresponds to a cell in the arrangement of these plates. Thus a sharp bound on the maximum combinatorial complexity of a single cell provides insight into the computational complexity of the motion planning problem and, in some cases, is the single most important quantity controlling the performance of a motion planning algorithm.

We also devote some attention to the problem of computing a single cell or a subset of cells in a three-dimensional arrangement of convex plates. A worst-case nearly optimal randomized algorithm for computing a single cell is described along with a less efficient procedure that constructs an arbitrary collection of cells. The justification of the single-cell algorithm relies on the so-called "Slicing Theorem" that states roughly that there exists a subdivision of a cell (or any collection of cells) in an arrangement of triangles into tetrahedra so that the number of resulting tetrahedra is not much larger than the combinatorial complexity of the cell(s) being triangulated. Finally, we examine a few special classes of arrangements for which better estimates on the maximum single-cell complexity can be deduced and where computing a cell or any collection of cells appears easier.

Part I

Geodesic Geometry in a Simple Polygon

Chapter 2

Geometry in a Simple Polygon

The analysis of geodesic Voronoi diagrams, which is the subject of the following two Chapters, relies heavily on some elementary properties of the internal geodesic metric, as formulated and proven below. In this metric, the distance between two points in a polygon is measured as the length of the shortest path that connects them and is contained in the polygon.

When working with a metric other than the familiar Euclidean distance, one immediately wonders about possible equivalents of traditional notions, such as a *straight line* and a *line segment*, an *angle* and a *triangle*, a *convex object* and a set of points in a *convex position*. Below we propose a way of defining these notions in a manner reminiscent of that in the standard Euclidean metric with the intention of "inheriting" most of the familiar Euclidean properties. However, one has to refrain from jumping to conclusions based on superficial similarity of these definitions and their intended equivalents in standard geometry. For example, it is natural to define a "line segment" from point x to point y as the shortest path connecting the two points and contained in our simple polygon (such a path is called a *geodesic*). The geodesic is indeed unique and has several other line-segment-like properties. However, the notion of "the line containing two given points" is more problematic, as there may in fact be infinitely many different maximal geodesics containing the two points, forcing us to either abandon the notion of a "line" or choose some canonical representative

geodesic passing through any pair of points. As in the latter case the choice is somewhat arbitrary, statements about such "lines" may have somewhat artificial form that critically depends on the precise definition of such a canonical geodesic. Unpleasant surprises of this sort have forced us to adopt a rather rigorous formal approach to the analysis of the geometry in a simple polygon, which the reader may find overly technical and pedantic. In fact, our experience with the geodesic metric seems to indicate that, apart from very elementary facts, theorems in the geodesic universe are exceptionally sensitive to the exact wording of their statements and that the precise phrasing of practically all definitions is unusually critical for the development of the results depending on them. Intuitively, the reason behind it is quite simple–geodesic metric admits a much larger variety of degeneracies than we are accustomed to in the Euclidean world and it appears that the only way to avoid those is to qualify theorem statements with additional seemingly unnecessary technical conditions. Two vivid examples of this phenomenon are the statements of Lemmas 2.2.3 and 2.2.4. In fact, one type of degeneracy will threaten our analysis so much that we will have to assume it away completely (we refer to this as the *general position assumption*, cf. Section 2.5).

This Chapter is devoted in its entirety to elementary facts of geodesic geometry. We begin with the definitions of geodesics and boundary geodesics, the latter roughly corresponding to infinite lines in the Euclidean geometry, though the analogy is somewhat misleading. The definitions of a *geodesic metric* and *geodesic direction* are followed by a discussion of some properties of the "shortest-path" map assigning to each point of the universe its distance from a fixed origin. We then proceed to define relative convexity and attempt to identify a "good" canonical extension for a geodesic (the sense in which we are not completely successful will become clear later, after we explore some properties of these extensions). Next, we present a rather detailed treatment of relatively convex sets (Section 2.3), formalizing in the process the notion of a counterclockwise ordering of points on the boundary of such a set. We then proceed to define the "far side" of a relatively convex set H from a fixed

point x–intuitively, it consists of the points of H that are hidden from x by H itself (Section 2.4). This notion is a useful tool for building relative convex hulls of larger sets from those of their components. Finally, we devote Section 2.5 to the general position assumption and its immediate consequences. The Chapter concludes with the definition of a bisector and a discussion of its properties in Section 2.6.

2.1 Preliminaries

The *universe* U is a compact region in the plane whose boundary ∂U is a simple n-sided polygon. S, the set of *sites,* is a collection of k points in U. A vertex of ∂U is called a *corner* and a segment of ∂U is a *wall.* A corner is *reflex* if the measure of its interior angle is more than π and *convex* if it is less than π. The *counterclockwise order* of points on ∂U is given by the traversal of ∂U that keeps the interior of U (locally) to its left. If x and y are distinct points of ∂U, then *boundary fragment* $\partial U[x, y]$ is the portion of ∂U counterclockwise from x to y inclusive. The symbol ∂ denotes the boundary (i.e., closure less interior) of a set relative to the whole plane, rather than to any proper subset of the plane. The terms "relative boundary" and "relative interior" used without any other qualification mean "relative to U."

A *polygonal path* is a simple path comprised of a sequence of line segments. If σ is a polygonal path, then a *link* of σ is a maximal segment of σ not containing any corners in its interior, the *size* of σ, denoted $\|\sigma\|$, is the number of its links, and the *length* of σ is the sum of their (Euclidean) lengths. A *polygonal region* is a compact, not necessarily connected, set whose boundary is the union of a finite number of line segments. We allow points as (degenerate) line segments, so for example a finite set of points is also a polygonal region.

2.2 Geodesics

For points $x, y \in U$, the *geodesic path* $g(x,y)$ is the shortest path in U connecting x and y. Such a shortest path is called simply a *geodesic*. This path is unique. In fact, it is a polygonal path with interior vertices only at reflex corners of ∂U [LP84]. We often consider $g(x,y)$ directed from x to y. Clearly an endpoint of a link of $g(x,y)$ is either x, y, or a corner. The first endpoint of the last link of $g(x,y)$ is the *anchor* of y with respect to x; it is either a reflex corner of ∂U or x itself.

We make heavy use of the fact that the intersection of any two geodesics is connected (and is itself a geodesic). This fact follows immediately from uniqueness of geodesics. Two polygonal paths *overlap* if they intersect in more than a single point.

The *geodesic distance* $d(x,y)$ between points x and y is the (Euclidean) length of $g(x,y)$. The geodesic distance is a metric; in particular, it is continuous in both x and y with respect to the Euclidean metric and satisfies the triangle inequality $d(x,z) \leq d(x,y)+d(y,z)$. Furthermore, by uniqueness of geodesics, $d(x,y)+d(y,z) = d(x,z)$ if and only if y lies on $g(x,z)$. We often write d_u for the function defined by $d_u(x) = d(u,x)$.

Lemma 2.2.1 (Pollack, Sharir, and Rote [PSR, Lemma 1]) *For any $u, v, w \in U$, d_u is a convex function on $g(v,w)$ with unique local minimum (possibly at v or w). In particular, for any $z \in g(v,w)$, $z \neq v, w$, one has $d_u(z) < \max\{d_u(v), d_u(w)\}$.*

The *geodesic direction* $\theta(x,y)$ from x to $y \neq x$ is the direction from x towards the anchor of x with respect to y. In other words, it is a unit vector based at x and directed along the first link of $g(x,y)$. At any point x not an anchor (of some other point) with respect to y, d_y is differentiable as a function of x and $\theta(x,y)$ is continuous as a function of both x and y. In fact, for fixed y, $\theta(x,y) = -\nabla_x d_y(x)$, the gradient of d_y with respect to x, evaluated at x. For H a closed subset of U, let $\theta(x,H)$ be $\{\theta(x,h) \mid h \in H, h \neq x\}$.

The *geodesic angle* $\angle xyz$ is the angle counterclockwise from $\theta(y,x)$ to $\theta(y,z)$. The

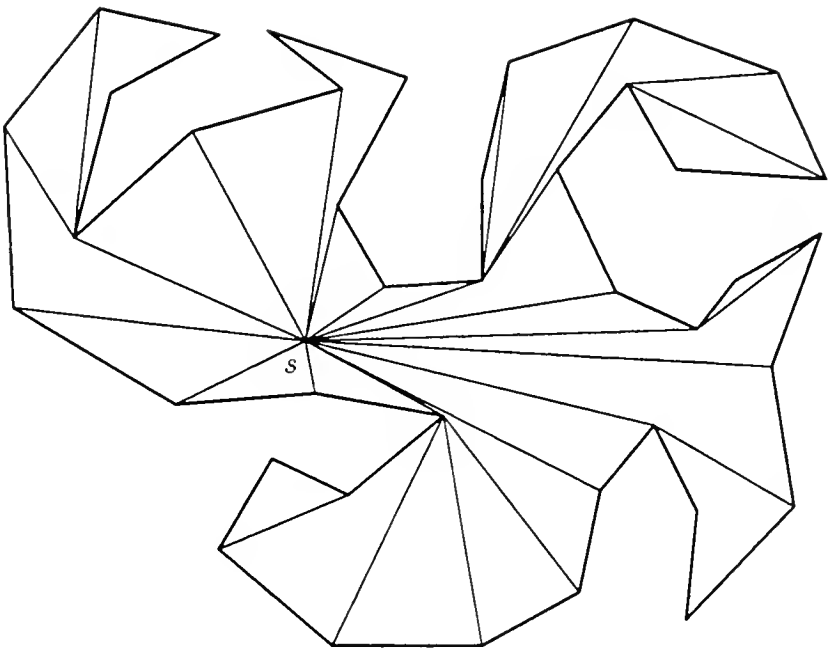


Figure 2.1: Shortest path tree $T(s)$ from s.

measure of $\angle xyz$ will be written $m\angle xyz$. The *angle between* $\theta(y,x)$ *and* $\theta(y,z)$ is the smaller of the two angles $\angle xyz$ and $\angle zyx$.

Lemma 2.2.2 (Pollack, Sharir, and Rote [PSR, Corollary 2]) *If* $x, y, z \in U$, $y \neq x, z$, *and the measure of the angle between* $\theta(y,x)$ *and* $\theta(y,z)$ *is at least* $\pi/2$, *then* $d(x,z) > \max\{d(x,y), d(y,z)\}$.

The *shortest path tree* from s, $T(s)$, is the union of the sets of links of $g(s,y)$ taken over all corners y of ∂U (Figure 2.1). It has $n-1$ or n links, depending on whether or not s itself is a corner of U [GHL*87].

Let $P_a(s)$ be the set of points in U that have anchor a with respect to s. The *shortest path partition* of U from s is the collection $\{P_a(s) \mid P_a(s) \neq \emptyset\}$ (refer to Figure 2.2). It is a planar polygonal subdivision of U; it can be computed and in fact triangulated in linear time given a triangulation of U [GHL*87]. We can describe

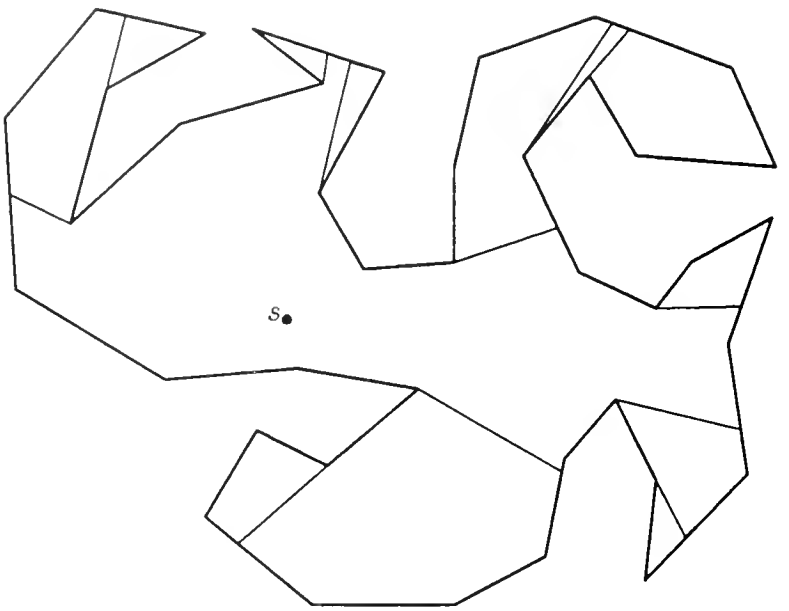


Figure 2.2: Shortest path partition of U from s.

the bounding edges of the shortest path partition as follows. Suppose $P_a(s)$ is not empty, where $a \neq s$. Let ab be the first link of the geodesic $g(a,s)$ (clearly ab is the second link of all geodesics $g(x,s)$ for $x \in P_a(s)$). Since $P_a(s)$ is not empty and a is a reflex corner of ∂U, we can extend link ab past a into U. First suppose that neither wall of ∂U incident to a overlaps this extension. Let y be the first point past a of the extension so that $y \in \partial U$. Then segment ay is the *shortest-path partition edge* (from s with anchor a), denoted $p_a(s)$. Now suppose some wall of ∂U incident to a overlaps the extension of segment ab; then we simply define $p_a(s)$ to be this wall. It can be checked that the boundary of a cell of the shortest path partition consists of an alternating sequence of shortest path partition edges and sections of ∂U. A convenient triangulation of the shortest path partition is obtained by further subdividing U along the links of $T(s)$ [GHL*87].

A set $A \subseteq U$ is *relatively convex* (with respect to U) if $g(x,y) \subseteq A$ whenever $x, y \in A$; the *relative convex hull* of set $F \subseteq U$, denoted $R(F)$, is the smallest

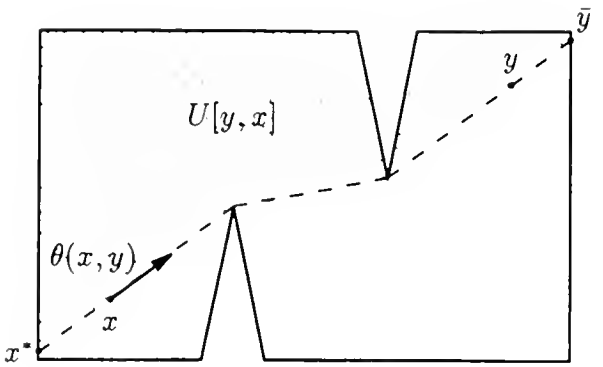


Figure 2.3: Geodesic $g(x,y)$, its shadow $\bar{y}$ and foreshadow x^* (they are unique in this case), the corresponding boundary geodesic $\bar{g}(x,y) = g(x^*,\bar{y})$, and geodesic direction $\theta(x,y)$. $U[y,x]$ is shaded.

relatively convex set containing F (that is, the intersection of all relatively convex sets containing F) [Tou86]. Notice that, by uniqueness of geodesics, the intersection of two relatively convex sets is relatively convex. Relatively convex sets are discussed in detail in the next Section. A set is *degenerate* if it is contained in a single geodesic.

If x and y are distinct points of U, then a *shadow* of $g(x,y)$ is a point $\bar{y} \in \partial U$ so that segment $y\bar{y}$ extends the last link of $g(x,y)$ while staying in U. Clearly y is a shadow of $g(x,y)$ only if $y \in \partial U$. Shadows are not unique, indeed it is possible that every point on a subsegment of a wall is a shadow of $g(x,y)$. Similarly, a *foreshadow* of $g(x,y)$ is a shadow of $g(y,x)$, i.e., a point $x^* \in \partial U$ lying on a segment contained in U extending the first link of $g(x,y)$ backwards. For an illustration of these definitions, see Figure 2.3.

A *boundary geodesic* is a geodesic connecting two distinct points of ∂U. Let x^* and $\bar{y}$ be the foreshadow closest to x and shadow closest to y of $g(x,y)$, respectively. Then $\bar{g}(x,y)$ denotes the boundary geodesic $g(x^*,\bar{y})$. Notice that $\bar{g}(x,y)$ coincides with $g(x,y)$ if and only if both x and y lie on ∂U. Geodesic $\bar{g}(x,y)$ splits U into two simply connected polygonal regions $U[x,y]$ and $U[y,x]$ with disjoint interiors; $\partial(U[x,y])$ is

$\partial U[x^*, \bar{y}] \cup g(\bar{y}, x^*)$ and $\partial(U[y,x])$ is $g(x^*, \bar{y}) \cup \partial U[\bar{y}, x^*]$. (Note that $\partial(U[x,y])$ is distinct from $\partial U[x,y]$.) $U[y,x]$ is shaded in Figure 2.3. Intuitively, $U[x,y]$ contains points lying on or to the right of $g(x^*, \bar{y})$, while points on the geodesic or to the left of it constitute $U[y,x]$. Notice that $\bar{g}(x,y)$ is exactly the common boundary of $U[x,y]$ and $U[y,x]$; hence any geodesic from a point in $U[x,y]$ to a point in $U[y,x]$ must intersect $\bar{g}(x,y)$. Both $U[x,y]$ and $U[y,x]$ are relatively convex, since a geodesic connecting two points of, say, $U[x,y]$ must have connected intersection with $\bar{g}(x,y)$. The following two Lemmas are technical results extensively used in subsequent analysis. The reader is encouraged to consider the statements in Euclidean geometry that are mimicked below.

Lemma 2.2.3 *Suppose* $u, v \in \partial U$, $u \neq v$, $w \in g(u,v)$, $x \in U[u,v]$, *and* $w \neq x$.

(1) If $x \notin g(u,v)$, *then* any *shadow* $\bar{x}$ *of* $g(w,x)$ *lies in* $\partial U[u,v]$.

(2) If $x \in g(u,v)$, *then* some *shadow* $\bar{x}$ *of* $g(w,x)$ *lies in* $\partial U[u,v]$.

Proof: (1) Suppose $x \notin g(u,v)$. Then $g(w,\bar{x})$ cannot intersect $\bar{g}(u,v)$ again after x, so $\bar{x} \in \partial U[u,v]$.

(2) Suppose $x \in g(u,v)$; without loss of generality assume w, x, v are in that order along $g(u,v)$. We can choose $\bar{x} = v$ unless $g(u,v)$ bends at or after x. If $g(u,v)$ bends right at some point c at or after x, then since $U[u,v]$ lies locally to the right of $g(u,v)$, c must be a reflex corner of $\partial U[u,v]$, and we can choose $\bar{x} = c$. If $\bar{g}(u,v)$ bends left at some point c then the straight-line continuation of $g(w,c)$ at c enters the interior of $U[u,v]$ and thus will not intersect $g(u,v)$ again. Hence we can choose $\bar{x}$ to be any shadow of $g(w,c)$ distinct from c. □

Lemma 2.2.4 *Suppose* $u, v \in \partial U$, $u \neq v$, $w \in g(u,v)$, $x \in U[u,v]$, $w \neq x$, *and* $\bar{w}$ *the closest shadow of* $g(x,w)$.

(1) If $w \notin \partial U$ *and* $g(x,w)$ *does not overlap* $g(u,v)$, *then* $\bar{w} \in \partial U[v,u]$.

(2) If v is the closest shadow of $g(u,w)$ and $g(u,w)$ is not an initial portion of $g(u,x)$ (i.e. $w \notin g(u,x)$), then $\bar{w} \in \partial U[v,u]$.

Proof: (1) If $g(x,w)$ does not overlap $g(u,v)$, then $\theta(w,x)$ cannot be $\theta(w,u)$ or $\theta(w,v)$. Since $w \notin \partial U$, $\bar{w} \neq w$ and $\theta(w,\bar{w}) = -\theta(w,x)$ must enter the interior of $U[v,u]$ at w. As $g(x,w)$ cannot intersect $g(u,v)$ again, $\bar{w} \in \partial U[v,u]$.

(2) The statement is trivial if $w \in \partial U$. If not, then $w \neq \bar{w}$. We might have $\theta(w,x) = \theta(w,u)$, in which case $\bar{w} = v$. We cannot have $\theta(w,x) = -\theta(w,u)$, else $w \in g(u,x)$. Otherwise $\theta(w,\bar{w})$ must enter the interior of $U[v,u]$ and as before $\bar{w} \in \partial U[v,u]$. □

Boundary geodesic $g(x,y)$ *separates* points a and b if $a \in U[x,y]$ and $b \in U[y,x]$, or vice versa. Similarly, boundary geodesic $g(x,y)$ *separates* sets A and B if $A \subseteq U[x,y]$ and $B \subseteq U[y,x]$, or vice versa. Note that separation does not imply disjointness, indeed if (degenerate) set A is contained in boundary geodesic $g(x,y)$, then $g(x,y)$ separates A from itself.

For a compact set $F \subseteq U$ and $z \in U$, let $rad(z,F) = \sup_{x \in F} d_z(x)$. The *(geodesic) center* of F is the point z that minimizes $rad(z,F)$. Pollack, Sharir, and Rote[PSR] show that the center of the set of vertices of U (or, as a matter of fact, of all of U) is unique. In fact, with minor modifications, their proof demonstrates the uniqueness of the center of any compact set $F \subseteq U$.

For points $x,y,z \in U$ we define the *geodesic triangle* Δxyz as follows [PSR]. If $\{x,y,z\}$ is degenerate, then Δxyz is the smallest geodesic containing x, y, and z. Otherwise we can choose points x', y', and z' so that x' is the point at which $g(x,y)$ and $g(x,z)$ diverge, and similarly for y' and z'. Refer to Figure 2.4. Then the circuit $g(x',y')$, $g(y',z')$, $g(z',x')$ is a simple (non self-touching) polygon $\Delta x'y'z'$. We define Δxyz to be the union of $g(x,x')$, $g(y,y')$, $g(z,z')$, and $\Delta x'y'z'$ together with its interior. All of the interior angles of $\Delta x'y'z'$ are reflex except at x', y', z' [PSR]. The geodesic triangle is a special case of the relative convex hull of a finite set of points discussed in the next Section. The proof of the following Lemma is in the same spirit as the proofs of Lemmas 2.2.3 and 2.2.4.

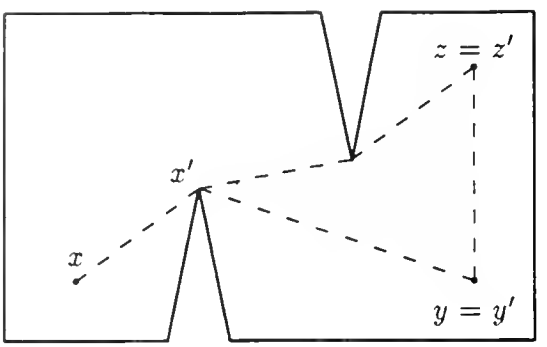


Figure 2.4: Geodesic triangle Δxyz.

Lemma 2.2.5 (Triangle Lemma) *Suppose $\{x, y, z\}$ is not degenerate. Let $\bar{y}$ and $\bar{z}$ be shadows of $g(x, y)$ and $g(x, z)$ respectively. Assume $z \in U[y, x]$.*

(1) If $u \in \Delta xyz$ and $u \notin g(x, y) \cap g(x, z)$ then there is a shadow $\bar{u}$ of $g(x, u)$ so that $\bar{u} \in \partial U[\bar{y}, \bar{z}]$ and $g(x, \bar{u})$ intersects $g(y, z)$.

(2) If u is in the interior of Δxyz, then for any shadow $\bar{u}$ of $g(x, u)$, $\bar{u} \in \partial U[\bar{y}, \bar{z}]$ and $g(x, \bar{u})$ intersects $g(y, z)$.

(3) If $u \in U[y, x] \cap U[x, z] \cap U[y, z]$, then $g(x, u)$ intersects $g(y, z)$ and there is a shadow $\bar{u}$ of $g(x, u)$ in $\partial U[\bar{y}, \bar{z}]$.

2.3 Relatively Convex Sets

This Section develops properties of relatively convex sets. Lemma 2.3.4 constitutes the main result. It states that the "extreme" elements of a set F can be ordered so that the relative convex hull of F is the intersection of all "cones" defined by consecutive triples of extreme elements. An immediate consequence of Lemma 2.3.4 is a decomposition of a relatively convex set into a collection of simple polygons and connecting geodesics. Also, the order of extreme elements extends to a natural notion

of a traversal of the boundary of a relatively convex set. This ordering has a number of useful consequences, given in Lemmas 2.3.6 through 2.3.9.

Lemma 2.3.1 *Any relatively convex set R is simply connected.*

Proof: We show that the region enclosed by a simple cycle γ in R lies completely in R. Suppose $x \in \gamma$, y is in the interior of γ, and $\bar{y}$ is a shadow of $g(x, y)$. Since $y\bar{y}$ connects a point inside γ to a point on or outside γ, there is a point w in the intersection of $y\bar{y}$ and γ. Since R is relatively convex, $g(x, w) \subseteq R$, so $y \in R$. □

Let F be a nonempty polygonal region contained in U. (Recall that polygonal regions need not be connected; in particular, F may be a finite set of points.) For $x \in U$, we define $span(x, F)$ to be the smallest superset Γ of $\theta(x, F)$ with the property that the angle between two non-opposite directions $\alpha, \alpha' \in \theta(x, F)$ is included in Γ whenever this angle is contained in U locally near x. It is easy to see that, if $x \in U$ is not a reflex corner of ∂U, then either $span(x, F)$ has a single component or $span(x, F)$ consists of two opposite directions. If x is a reflex corner of ∂U, then $span(x, F)$ may have two components that are not opposite directions; in fact one or both components may have positive measure. Since F is a polygonal region, it is easy to check that the endpoint of any component of $span(x, F)$ is $\theta(x, y)$ for some $y \in F$. We show below (Lemma 2.3.3) that $span(x, F) = \theta(x, R(F))$, which is an equivalent definition of *span*. The less intuitive definition was chosen primarily to facilitate the proof of the next Lemma.

If $span(x, F)$ consists of a single component of measure less than 2π, then x is an *exterior point* of F. Notice that $span(x, F)$ can have measure exceeding π but less than 2π only if x is a reflex corner of ∂U. If $span(x, F)$ consists of a single component of measure less than π and $x \in F$ then x is an *extreme point* of F. If $span(x, F)$ consists of two connected components, then x is a *thin* point of F. In Figure 2.5, for $F = \{1, 2, 3, 4, 5\}$, point 5 and all points of ∂U except a are exterior (but none are extreme), points 1, 2, and 3 are extreme points, and every point on $g(1, a)$ except 1 is a thin point.

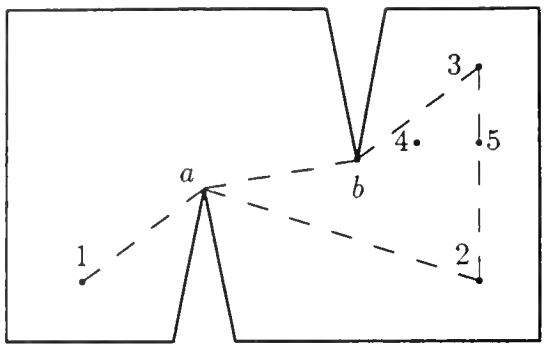


Figure 2.5: Relative convex hull of $\{1, 2, 3, 4, 5\}$.

For the remainder of this Section, $F \subseteq U$ is a finite set of points containing at least two elements. Set F is *extreme* if every element of it is extreme. In Figure 2.5, $\{1, 2, 3\}$ is extreme.

Let x be an exterior point of F or a point of ∂U. If $x \in \partial U$ define $opp_F(x)$ to be x, otherwise let $opp_F(x)$ be the first point of ∂U intersected by the ray with endpoint x directed opposite the bisector of $span(x, F)$. The *clockwise extreme point* of F from x, denoted $r(x)$, is the point $f \in F$ so that there is no geodesic extending $g(x, f)$ to a point $f' \in F$, and among all such points the closest shadow of $g(x, f)$ is as clockwise as possible in $\partial U - \{opp_F(x)\}$. Similarly define the *counterclockwise extreme point* of F from x, denoted $l(x)$. For example, in Figure 2.5, $l(2) = 1$, $r(2) = 3$, $l(a) = 1$, $r(a) = 2$. There are two subtleties to these definitions. First, r and l depend upon the set F, but except in the proof of Lemma 2.4.3, we leave this dependence implicit. Second, there are two distinct (though overlapping) cases in the definition: either x is exterior (and $span(x, F)$ consists of a single component) or $x \in \partial U$ (and even if $span(x, F)$ consists of two components, there is still a natural way to define $r(x)$ and $l(x)$). Notice there is no natural definition of $r(x)$ and $l(x)$ if $span(x, F)$ has measure 2π or if $span(x, F)$ has two components and $x \notin \partial U$. The following Lemma describes properties of functions r and l; its proof is quite technical and detailed.

Lemma 2.3.2 *Let x be an exterior point of F or a point of ∂U, $r = r(x)$, $l = l(x)$, $\bar{r}$ the closest shadow of $g(x, r)$, and $\bar{l}$ the closest shadow of $g(x, l)$.*

1. *If x is exterior then $span(x, F) = \angle rxl$. If $x \in \partial U$ or $m\angle rxl \neq \pi$ then $F \subseteq U[r, x] \cap U[x, l]$; if $m\angle rxl = \pi$ then $F \subseteq U[r, l]$.*

2. *If $r \neq l$, then $r\bar{r}$ and $l\bar{l}$ are disjoint; if also $x \in \partial U$ then x, $\bar{r}$, $\bar{l}$ are in that counterclockwise order around ∂U.*

3. *Both r and l are extreme points of F.*

4. *If x is an extreme point of F, then $r(l(x)) = l(r(x)) = x$.*

5. *Let $x \in \partial U$ or $m\angle rxl < \pi$. Then $f = r(x)$ if and only if $f \in F$, $F \subseteq U[f, x]$, and $g(x, f)$ cannot be extended to $g(x, f')$ for any other $f' \in F$.*

Proof:

1. We must always have $span(x, F) \subseteq \angle rxl$ by definition of $span(x, F)$ and of points r and l; certainly $\theta(x, r), \theta(x, l) \in span(x, F)$. If x is exterior then $span(x, F)$ is connected, so we must have $\angle rxl = span(x, F)$. For the second statement, first suppose $x \in \partial U$ or $m\angle rxl \neq \pi$. Let x^* be the closest foreshadow of $g(x, r)$. We claim $\bar{l} \in \partial U[\bar{r}, x^*]$: this follows immediately from the definition of r and l if $m\angle rxl < \pi$, $x \in \partial U$, or $m\angle rxl > \pi$ (since in the last case necessarily $x \in \partial U$). For any $f \in F$, either $f \in g(x, r)$, $f \in g(x, l)$, or the closest shadow of $g(x, f)$ lies in $\partial U[\bar{r}, \bar{l}] \subseteq \partial U[\bar{r}, x^*]$, so $f \in U[r, x]$, and $F \subseteq U[r, x]$. Similarly $F \subseteq U[x, l]$. The argument that $m\angle rxl = \pi$ implies $F \subseteq U[r, l]$ is similar. For an illustration of this somewhat unintuitive case, refer to Figure 2.5, placing x at any point of segment ab other a or b. For such a point x, $r(x) = 1$, $l(x) = 3$, and indeed $F = 1, 2, 3, 4, 5 \subset U[1, 3]$. On the other hand, $F \not\subseteq U[x, 3] = U[a, 3]$.

2. Suppose $l \neq r$. If $l\bar{l}$ and $r\bar{r}$ met, either l would lie on $g(x, r)$ or vice versa, contrary to the definition of r, l. The ordering of x, $\bar{r}$, and $\bar{l}$ follows immediately by definition.

3. Referring to part 1 of the Lemma, we assume $F \subseteq U[r,x]$, otherwise a similar argument works using $U[r,l]$. To show r extreme, we need to demonstrate that $span(r,F)$ is connected and has measure less than π. Whether or not $r \in \partial U$, $\theta(r,U[r,x])$ is connected. As $F \subseteq U[r,x]$, $span(r,F) \subseteq \theta(r,U[r,x])$. Hence it suffices to show that for $f \in F$, $\theta(r,f)$ lies in the angle from $\theta(r,x)$ clockwise to but not including $-\theta(r,x)$. Now $\theta(r,f)$ cannot be $-\theta(r,x)$ else $g(x,f)$ would extend $g(x,r)$. Also $\theta(r,f)$ can be clockwise of $-\theta(r,x)$ only in the case that r is a reflex corner of ∂U so that the exterior of U is locally contained in the geodesic angle $\angle xrf$ and $m\angle xrf < \pi$; again this is impossible because $g(x,f)$ would extend $g(x,r)$.

4. By (1), $F \subseteq U[r,x]$. For any $f \in F$ not appearing on $g(x,r)$, f is in the relative interior of $U[r,x]$. Thus, by Lemma 2.2.3(1), the closest shadow of $g(r,f)$ lies on $\partial U[\bar{r},x^*]$, where x^* is the closest foreshadow of $g(x,r)$. This implies $x = l(r) = l(r(x))$ by definition of counterclockwise extreme point.

5. Clearly there can be at most one point f in F satisfying "$F \subseteq U[f,x]$ and $g(x,f)$ cannot be extended to $g(x,f')$ for any $f' \in F$". Since $f = r(x)$ is one such point, the claim follows.

□

Lemma 2.3.3 *For $x \in U$, $span(x,F) = \theta(x,R(F))$.*

Proof: Suppose $y,z \in F$ are such that $0 < m\angle yxz < \pi$ and $\angle yxz$ is (locally around x) contained in U. By examining Δxyz and using $g(y,z) \subseteq R(F)$ we see that $\angle yxz \subseteq \theta(x,R(F))$. Hence $span(x,F) \subseteq \theta(x,R(F))$.

If $span(x,F)$ has measure 2π, then it must be that $span(x,F) = \theta(x,R(F))$. Suppose x is an exterior point of F; let $r = r(x)$ and $l = l(x)$. If $m\angle rxl \neq \pi$ then $R(F) \subseteq U[r,x] \cap U[x,l]$ since $F \subseteq U[r,x] \cap U[x,l]$ by Lemma 2.3.2(1), and $U[r,x]$ and $U[x,l]$ are relatively convex. Hence $\theta(x,R(F)) \subseteq \theta(x,U[r,x] \cap U[x,l]) = \angle rxl = span(x,F)$, where the first equality follows by definition and the second by

Lemma 2.3.2(1). If $m\angle rxl = \pi$ then $R(F) \subseteq U[r,l]$, so $\theta(x, R(F)) \subseteq \theta(x, U[r,l]) = \angle rxl = span(x,F)$.

Finally, if x is a thin point of F, we can find a segment through x splitting U into two simple polygons, each containing one of the components of $span(x,F)$. Let F_1 and F_2 be the intersections of these two polygons with F, and let H_1, H_2 be their respective convex hulls (relative to U). It is easy to verify that x is an exterior point of both F_1 and F_2 and thus, by the first part of the proof,

$$span(x,F_1) \cup span(x,F_2) = \theta(x,H_1) \cup \theta(x,H_2) \subseteq \theta(x,R(F)). \qquad (2.1)$$

Since trivially $span(x,F_1) \cup span(x,F_2) \subseteq span(x,F)$, it is sufficient to show that the inclusion in (2.1) is an equality. Suppose there is a point $y \in R(F) - \{x\}$ such that $\theta(x,y) \in \theta(x,R(F)) - \theta(x,H_1) \cup \theta(x,H_2)$. As $y \in R(F) = R(H_1 \cup H_2)$, there are two points, $z_1 \in H_1$ and $z_2 \in H_2$ so that $y \in g(z_1,z_2)$. Notice that $x \notin g(z_1,z_2)$, hence $\Delta x z_1 z_2 \subseteq R(F)$ is not degenerate and $\theta(x, g(z_1,z_2)) \subseteq span(x,F)$ is the angle between $\theta(x,z_1)$ and $\theta(x,z_2)$, contradicting the assumption that x is a thin point and $\theta(x,z_1)$ and $\theta(x,z_2)$ lie in different components of $span(x,F)$. □

It is immediate that any two sets with the same relative convex hull have the same extreme, exterior, and thin points.

The next Lemma is the main result of this Section. To state it we need the notion of a geodesic cone. Suppose $m\angle xyz < \pi$ and $z \in U[x,y]$. Let σ be the segment from y to the closest foreshadow y^* of $g(y,x)$, open at y and closed at y^*. The *geodesic cone* $U[x,y,z]$ is $U[x,y] \cap U[y,z] - \sigma$. See Figure 2.6. Notice that σ intersects $U[x,y] \cap U[y,z]$ only if σ is also a foreshadow segment of $g(y,z)$. It is easily checked that $U[x,y,z]$ is relatively convex and that the boundary of $U[x,y,z]$ consists of $g(\bar{z},y)$, $g(y,\bar{x})$, and $\partial U[\bar{x},\bar{z}]$, where $\bar{x}$ and $\bar{z}$ are the closest shadows of $g(y,x)$ and $g(y,z)$, respectively. Furthermore, Lemma 2.3.2(1) implies that, if f is an extreme point of F, then $F \subseteq U[r(f),f,l(f)]$.

Lemma 2.3.4 *For a finite set F, $H = R(F)$ is a simply connected polygonal region. The extreme elements of F can be labelled f_1, $f_2 = r(f_1)$, ..., $f_0 = f_m = r(f_{m-1})$.*

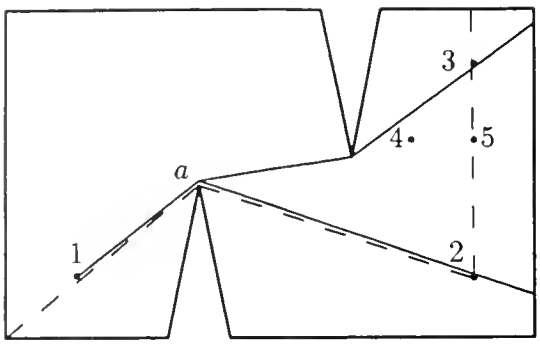


Figure 2.6: Geodesic cones $U[2,1,3]$ (solid) and $U[3,2,1]$ (dashed).

$f_1 = f_{m+1} = r(f_m)$ so that $H = \bigcap_{i=1}^{m} U[f_{i+1}, f_i, f_{i-1}]$ and $\partial H = \bigcup_{i=1}^{m} g(f_i, f_{i+1})$.

Proof: F must contain some extreme element, since $r(u)$ is extreme for any $u \in \partial U$. Let f_1 be an extreme element of F, and consider the sequence $f_2 = r(f_1)$, $f_3 = r(f_2)$, ... Since F is finite and $l \circ r$ is the identity (Lemma 2.3.2(4)), it must be that $f_{m+1} = f_1$ for some m with $f_1, ..., f_m$ distinct. If, for some i, $f_{i-1} = f_{i+1}$, F is degenerate, $m = 2$ and the Lemma follows easily. We thus assume that the points in each triple f_{i-1}, f_i, f_{i+1} are distinct, so that F is nondegenerate and $m > 2$.

Let $I = \bigcap_{i=1}^{m} U[f_{i+1}, f_i, f_{i-1}]$. Our goal is to show $\partial I = \bigcup_{i=1}^{m} g(f_i, f_{i+1})$. Let $g(f_i, f_{i+1})$ have closest foreshadow s_i and closest shadow t_i. A schematic view of the situation is given in Figure 2.7. Notice that σ intersects $U[x,y] \cap U[y,z]$ only if σ is also a indicates the relative positions of these points, to be justified below. Then $\partial(U[f_{i+1}, f_i, f_{i-1}]) = g(f_i, t_i) \cup g(s_{i-1}, f_i) \cup \partial U[t_i, s_{i-1}]$. By Lemma 2.3.2(1) we have for $j = 1, \ldots, m$, $f_{j-1} \in U[f_{j+1}, f_j] = U[t_j, s_j]$, so Lemmas 2.2.3 and 2.2.4 guarantee that $s_{j-1}, s_j, t_{j-1}, t_j$ lie in that counterclockwise order on ∂U (possibly $s_j = t_{j-1}$). Consequently, $\bigcup_{j=1}^{m} \partial U[s_j, t_j] = \partial U$. Hence $\bigcap_{j=1}^{m} \partial U[t_j, s_{j-1}] \subseteq \bigcap_{j=1}^{m} \partial U[t_j, s_j] \subseteq \{s_1, \ldots, s_m, t_1, \ldots, t_m\}$. We now claim that $\partial U[t_i, s_{i-1}] \cap I \subseteq \bigcup_{j=1}^{m} g(s_j, t_j)$. Suppose $x \in \partial U[t_i, s_{i-1}] \cap I$; either $x \in \bigcap_{j=1}^{m} \partial U[t_j, s_{j-1}]$, in which case the claim is immediate, or $x \notin \bigcap_{j=1}^{m} \partial U[t_j, s_{j-1}]$. In the second case, $x \notin \partial U[t_j, s_{j-1}]$ for some j, but $x \in U[f_{j+1}, f_j, f_{j-1}]$ as $x \in I$, and since $x \in \partial U$, it must be that $x \in \partial(U[f_{j+1}, f_j, f_{j-1}])$,

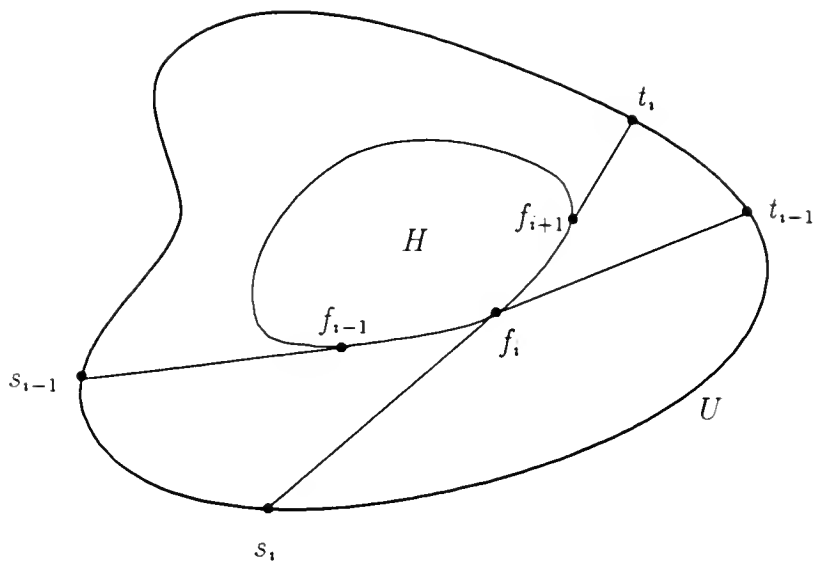


Figure 2.7: Illustration to the proof of Lemma 2.3.4

so $x \in g(f_j, t_j) \cup g(s_{j-1}, f_j)$, establishing the claim. We now show $\partial I = \bigcup_{i=1}^{m} g(f_i, f_{i+1})$. By elementary topology,

$$\begin{aligned}
\partial I &= \bigcup_{i=1}^{m} I \cap \partial(U[f_{i+1}, f_i, f_{i-1}]) \\
&= \bigcup_{i=1}^{m}((g(f_i, t_i) \cup g(f_i, s_{i-1})) \cap I) \cup \bigcup_{i=1}^{m}(\partial U[t_i, s_{i-1}] \cap I) \\
&= \bigcup_{i=1}^{m} g(s_i, t_i) \cap I.
\end{aligned}$$

Since segments $s_i f_i$ and $t_{i-1} f_i$ are in $U[f_{i+1}, f_i, f_{i-1}]$ only if $s_i = t_{i-1} = f_i \in \partial U$, we have $\partial I = \bigcup_{i=1}^{m} g(f_i, f_{i+1})$.

For each i, we have $H \subseteq U[f_{i+1}, f_i, f_{i-1}]$ since $U[f_{i+1}, f_i, f_{i-1}]$ is relatively convex and contains F. Hence $H \subseteq I$. Also $\partial I \subseteq H$ since each geodesic is contained in H: as H is simply connected, $H = I$. Every extreme element f of F must be in H and in fact on ∂H, for f is on the boundary of $U[r(f), f, l(f)]$. Since no extreme element can lie on a geodesic between two other extreme elements, each extreme element is

equal to f_i for some i. □

Since $R(F)$ is a connected polygonal region, it can be decomposed into *plateaus* and *bridges*: a *non-degenerate plateau* is a maximal closed two-dimensional region whose boundary is a simple polygon and a bridge is either a maximal polygonal path of positive length not containing any plateau points in its interior or a single point shared between two plateaus. We require that both endpoints of a bridge lie in a plateau, thus introducing *a degenerate plateau* at the endpoint of any positive-length bridge that does not end in a point of a non-degenerate plateau. In Figure 2.5, point 1 and geodesic triangle $\Delta 23a$ are plateaus (degenerate and non-degenerate, respectively); segment $1a$ is a bridge. Each plateau is relatively convex and each bridge is a geodesic. Since $R(F)$ is simply connected, this decomposition forms a tree, with plateaus taken as nodes and bridges as edges. An extreme point x of $R(F)$ is a convex vertex of a plateau (if $span(x, F)$ has positive measure) or is a plateau itself (if $span(x, F)$ consists of a single direction). A thin point x of $R(F)$ is an interior point of a bridge (if $span(x, F)$ consists of exactly two directions), an endpoint of a bridge of positive length (if $span(x, F)$ consists of two components only one of which has positive measure), or a bridge by itself (if $span(x, F)$ consists of two components both of positive measure).

For F labelled as in Lemma 2.3.4 and $H = R(F)$, we define the *counterclockwise traversal* of ∂H to be the circuit $g(f_1, f_2), \ldots, g(f_m, f_1)$. The counterclockwise traversal visits every thin point twice and other points once. Hence for x, y not thin points we can unambiguously define $\partial H[x, y]$ to be the section of the counterclockwise traversal of ∂H from x to y. Similarly a collection H' of non-thin points of ∂H has an unambiguous counterclockwise ordering along ∂H. If F is extreme, then the counterclockwise ordering covers every point of F. Even if the sequence of points $x_1, \ldots, x_k$ on ∂H includes thin points, it is still meaningful to say that they appear in counterclockwise order, if they are visited in that order in a single counterclockwise traversal of ∂H. Of course, another order differing in the position of thin points may be consistent with a counterclockwise traversal as well.

Corollary 2.3.5 *The center of F is the center of $R(F)$.*

Proof: It is sufficient to show that for any $x \in U$, $rad(x, F) = rad(x, R(F))$. A point of $H = R(F)$ maximally distant from x must lie on ∂H. By the previous Lemma any point on ∂H lies on a geodesic connecting two points of F; hence by Lemma 2.2.1, the maximally distant point must be an endpoint of the geodesic, that is, a point of F. Thus a point of H furthest from x is necessarily a point of F and $rad(x, H) = rad(x, F)$. □

Lemma 2.3.6 *If f, f' are extreme points of F, then every extreme point counterclockwise from f to f' is in $U[f, f']$, and all but f and f' lie in the relative interior of $U[f, f']$.*

Proof: The proof is by induction on the extreme points of F lying in counterclockwise order around F from f to f'. If $r(f) = f'$, the claim is vacuously true. If $r(f) \neq f'$, then since f' must lie in the relative interior of $U[r(f), f]$, $r(f)$ must lie in the relative interior of $U[f, f']$. Now suppose f'' is counterclockwise of f before f' and $r(f'') \neq f'$. By inductive assumption, f'' is in the relative interior of $U[f, f']$. Again we have f' in the relative interior of $U[r(f''), f'']$, so $r(f'')$ must be in the relative interior of $U[f'', f']$. Now $U[f'', f']$ is contained in $U[f, f']$ except possibly for a region bounded by segments $f's'$ and $f's''$ and $\partial U[s', s'']$, where s' and s'' are the closest shadows of $g(f, f')$ and $g(f'', f')$, respectively. But $g(f'', r(f''))$ cannot intersect segment $f's'$, since f' is extreme. Hence $r(f'') \in U[f, f']$. Since $r(f'')$ cannot lie on $\bar{g}(f, f')$ as f, f', $r(f'')$ are extreme and distinct, in fact $r(f'')$ is in the relative interior of $U[f, f']$. □

An immediate consequence of Lemma 2.3.6 is that counterclockwise order of extreme points is an absolute order, not depending on other extreme points: if a, b, c are extreme points of both sets A and B, then their order around A is the same as it is around B.

Lemma 2.3.7 *If points x, y, z, w occur in that order (not necessarily all distinct) in a counterclockwise traversal of the boundary of a relatively convex polygonal region R, then $g(x,z)$ intersects $g(y,w)$.*

Proof: By creating dummy plateaus if necessary, we can assume that x, y, z, and w each lie in a plateau. We proceed by induction on the number of plateaus in the decomposition of R. If there is only a single plateau, then R is a simple polygon and the claim is immediate. Otherwise there is some bridge $g(a,b)$ whose removal splits R into relatively convex polygonal regions R_1 and R_2. If all of x,y,z,w are contained in one of R_1 or R_2, the claim follows by induction. Otherwise, since x,y,z,w appear in that order in a counterclockwise traversal of the boundary of R, there are up to symmetry two cases: x,y,a in R_1 and z,w,b in R_2, or x,y,z,a in R_1 and b,w in R_2 (assuming that none of the four points lie in $g(a,b)-\{a,b\}$). In the first case $g(x,z)$ and $g(y,w)$ both contain $g(a,b)$. In the second case x,y,z,a appear in that order in a counterclockwise traversal of R_1, hence by induction hypothesis, $g(x,z)$ intersects $g(y,a)$. Since $g(y,a) \subseteq g(y,w)$, $g(x,z)$ must intersect $g(y,w)$. □

We say that segment $fx \subseteq U$ *connects* a relatively convex set R to ∂U if f is an extreme point of R, $x \in \partial U$, and fx intersects R only at f (possibly $f = x$ or some point of fx besides x lies on ∂U).

Lemma 2.3.8 (Connection Lemma) *Suppose $f_1,...,f_m$ are extreme points of relatively convex set R and segments f_ix_i are pairwise disjoint and connect R to ∂U. Then the order of the f_i's around ∂R is the same as the order of x_i's around ∂U.*

Proof: Suppose f_i, f_j, f_k are in counterclockwise order around ∂R; we show $x_j \in \partial U[x_i,x_k]$, i.e. x_i, x_j, x_k are in counterclockwise order around ∂U. By Lemma 2.3.6, f_j is in the relative interior of $U[f_i,f_k]$. Since $g(f_i,f_k) \subseteq R$ and f_jx_j meets R only at f_j, f_jx_j cannot intersect $g(f_i,f_k)$. Clearly f_j is in the region bounded by $\partial U[x_i,x_k]$, segment f_kx_k, $g(f_k,f_i)$, and segment f_ix_i. Since $x_j \in \partial U$ and f_jx_j does not intersect f_kx_k, $g(f_k,f_i)$ or f_ix_i, $x_j \in \partial U[x_i,x_k]$ we must have and $x_j \neq x_i$ and $x_j \neq x_k$. □

Lemma 2.3.9 *Mappings r and l restricted to ∂U preserve order.*

Proof: Suppose $u_1, u_2, u_3 \in \partial U$ are in that counterclockwise order and $r_1 = r(u_1)$, $r_2 = r(u_2)$, and $r_3 = r(u_3)$ are distinct. Let $\bar{r}_i$ be the closest shadow of $g(u_i, r_i)$.

We show that $\bar{r}_1, \bar{r}_2, \bar{r}_3$ lie in that counterclockwise order on ∂U; this follows from the claim that $\partial U[u_j, \bar{r}_j] \not\subseteq \partial U[u_i, \bar{r}_i]$ for all $i \neq j$. To establish this claim, suppose for the sake of contradiction that $\partial U[u_j, \bar{r}_j] \subseteq \partial U[u_i, \bar{r}_i]$ for some $i \neq j$. Then $r_j \in U[u_i, r_i]$ since $u_j, \bar{r}_j \in U[u_i, r_i]$. However, $r_j \in F \subseteq U[r_i, u_i]$ by Lemma 2.3.2(1), so $r_j \in g(u_i, \bar{r}_i)$. In fact $r_j \in g(u_i, r_i)$ since r_j cannot lie past r_i on $g(u_i, \bar{r}_i)$. If $r_j \in \partial U$ then $r_j = \bar{r}_j \in \partial U[u_i, \bar{r}_i]$, so either $r_i = r_j$ or $g(u_j, r_j)$ can be extended to $g(u_j, r_i)$, a contradiction. If $r_j \notin \partial U$, then neither $g(u_i, \bar{r}_i)$ nor $g(u_j, \bar{r}_j)$ can bend at r_j. Since $u_j, r_j \in U[u_i, r_i]$, the two geodesics must overlap in some link containing r_j. Furthermore the link must be traversed in the same direction in both geodesics, since u_j, u_i, $\bar{r}_i$, $\bar{r}_j$ appear in that counterclockwise order on ∂U. But then $g(u_j, r_j)$ could be extended to $g(u_j, r_i)$, a contradiction.

We claim that $r_i\bar{r}_i$ and $r_j\bar{r}_j$ are disjoint whenever $i \neq j$; the Lemma then follows immediately by the Connection Lemma. Suppose $r_i\bar{r}_i$ intersects $r_j\bar{r}_j$. The intersection must be a single point distinct from both r_i and r_j. Hence, either $r_j \notin U[r_i, u_i]$ or $r_i \notin U[r_j, u_j]$, contradicting Lemma 2.3.2(1). The claim for l follows by a symmetric argument. □

2.4 Far Sides

We consider the notion of the "far side" of a set F from a point. Informally, an extreme point $f \in F$ is on the far side of F from x if the geodesic $\bar{g}(x, f)$ leaves $R(F)$ after f. There are two main reasons for studying far sides. First, we use this notion to relate the order of extreme points of F with the order of points on ∂U (Corollary 2.4.2). Second, using far sides, we will be able to compute relative convex hulls of sets of a particular form efficiently (Lemma 2.4.3).

Let F be a finite set of points in U and $x \in \partial U$. The *far side* of F from x is

the set of all extreme points of F counterclockwise from $r(x)$ to $l(x)$, inclusive. Here $r(x)$ and $l(x)$ are the clockwise and counterclockwise extreme points of F from x, as before.

Lemma 2.4.1 *Suppose f is an extreme point of F, $x \in \partial U$, and $f \neq x$. Then the following are equivalent.*

1. *f is on the far side of F from x.*
2. *f is an extreme point of $F \cup \{x\}$.*
3. *$f\bar{f}$ connects $R(F)$ to ∂U, for any shadow $\bar{f}$ of $g(x,f)$, and there is no geodesic extending $g(x,f)$ to an extreme point f' of F distinct from f.*

Proof: If $x \in R(F)$ then all three conditions are automatically satisfied, so assume $x \notin R(F)$. Then x is an exterior point of F. Set $r = r(x)$ and $l = l(x)$. By Lemma 2.3.2(1) $f \in U[r,x] \cap U[x,l]$.

(1 $\Rightarrow$ 2) If f is on the far side of F from x, then $f \in U[r,l]$ by Lemma 2.3.6. If $f \neq r$ and $f \neq l$, then by the Triangle Lemma, $g(x,f)$ intersects $g(l,r)$ at some point x', where $x' \neq f$ since f is extreme. Hence $\theta(f,x) = \theta(f,x') \in \theta(f,g(r,l)) \subseteq span(f,F)$, and $span(f,F \cup \{x\}) = span(f,F)$, so f is extreme in $F \cup \{x\}$. If $f = r$, then by the definition of clockwise extreme point, $r(x)$ is unchanged if F is replaced with $F \cup \{x\}$, so r is extreme by Lemma 2.3.2(3). The case $f = l$ is similar.

(2 $\Rightarrow$ 3) Trivial.

(3 $\Rightarrow$ 1) We argue the contrapositive. If f is not on the far side of F, then by Lemma 2.3.6, $f \in U[l,r]$. Since f cannot lie on $\bar{g}(l,r)$, $f \in \Delta rxl$. If $f \in g(x,r)$ or $f \in g(x,l)$, then $g(x,f)$ can be extended to an extreme point of F distinct from f. Otherwise by the Triangle Lemma, for some shadow $\bar{f}$ of $g(x,f)$, $g(x,\bar{f})$ intersects $g(r,l)$ at a point distinct from f, thus $f\bar{f}$ cannot connect $R(F)$ to ∂U.

□

Corollary 2.4.2 *Suppose $\{f_i\}$ is a set of distinct points on the far side of F from x and, for each i, $\bar{f}_i$ is a shadow of $g(x, f_i)$. Then the order of $\bar{f}_i$ on ∂U is the same as the order of f_i on $\partial R(F)$.*

Proof: If $i \neq j$ then $g(x, f_i)$ and $g(x, f_j)$ must diverge before reaching f_i or f_j. Hence $f_i\bar{f}_i$ does not intersect $f_j\bar{f}_j$. The result then follows from the Connection Lemma. □

Lemma 2.4.3 *Suppose $u, v \in \partial U$, $u \neq v$, $F \subseteq U[v, u]$, $F \not\subseteq g(u, v)$, and $H = R(F)$. Then $\partial U[u, v]$, $g(v, r(v))$, $\partial H[r(v), l(u)]$, $g(l(u), u)$ (or $\partial U[u, v]$, $g(v, r(v))$, $g(l(u), u)$, if $r(v) = l(u)$) constitute a counterclockwise traversal of the boundary of $R(F \cup \partial U[u, v])$.*

Proof: In this proof, we write for example $r_H(x)$ and $l_H(x)$ for the clockwise and counterclockwise extreme points of H from x. This indicates the dependence upon H explicitly. Thus in the statement of the Lemma, $r(x)$ really refers to $r_F(x)$. We also assume $\partial U[u, v] \not\subseteq R(F)$ and $r(v) \neq l(u)$, hence two or more points of $\partial U[u, v]$ are extreme in $R(F \cup \partial U[u, v])$. If $\partial U[u, v] \subseteq R(F)$ or $r(v) = l(u)$, a similar and easier argument suffices.

Refer to Figure 2.8. Let G be $\{u, v\}$ together with the set of convex corners of $\partial U[u, v]$. Clearly $R(F \cup \partial U[u, v]) = R(F \cup G)$. We first show that $r_{F \cup G}(v) = r_F(v)$; we will use Lemma 2.3.2(5). Some point of F is in the relative interior of $U[v, u]$ by assumption; it is easy to check that in fact $r_F(v)$ must be in the relative interior of $U[v, u]$. Hence $F \cup \partial U[u, v] \subseteq U[r_F(v), v]$. Any geodesic extending $g(v, r_F(v))$ must stay in the relative interior of $U[v, u]$, hence must avoid $\partial U[u, v]$, and must also avoid any $f' \in F$ distinct from $r_F(v)$ since $r_F(v)$ is extreme in F. Hence by Lemma 2.3.2(5), $r_{F \cup G}(v) = r_F(v)$. By Lemma 2.3.2(3), $r_F(v)$ is extreme in $F \cup G$. By a similar argument, $l_{F \cup G}(u) = l_F(u)$ is extreme in $F \cup G$.

We claim $h = l_{F \cup G}(r_F(v))$ is the most clockwise point of G (ordered along ∂U) that is extreme in $F \cup G$: h is either v, the convex corner immediately clockwise of

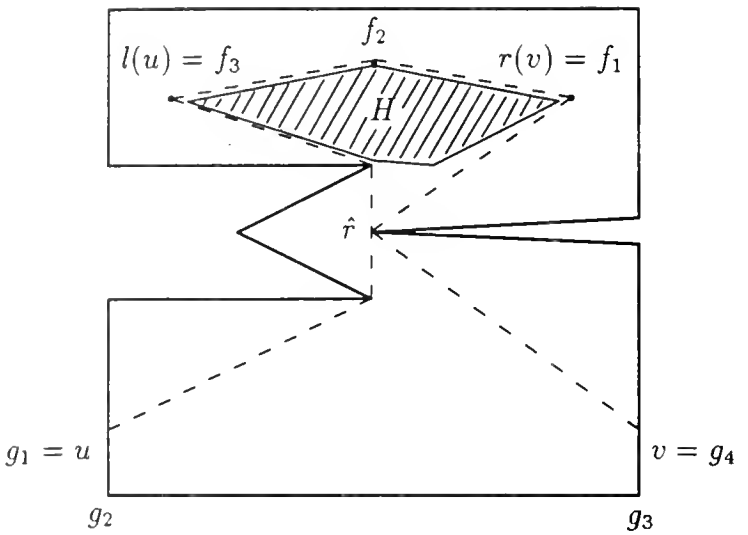


Figure 2.8: Dashes outline $R(F \cup \partial U[u,v])$.

v, or possibly u if $\partial U[u,v]$ contains no convex corners. We have $h \in U[r_F(v), v]$ and $v \in U[r_F(v), h]$. This is only possible if v, h and $r_F(v)$ lie on a common geodesic. Since h and $r_F(v)$ are extreme in $F \cup G$, they must be the endpoints of the geodesic. Hence either $v = h = l_{F\cup G}(r_F(v))$, or $g(r_F(v), v)$ extends to $g(r_F(v), h)$. Certainly $h \in U[r_F(v), v]$, so $v \in \partial U$ and $G \subseteq U[r_F(v), v]$ force $h \in \partial U[u,v]$; and $g(r_F(v), h)$ not extendible to another point of $F \cup G$ forces h to be u or the convex corner of G immediately clockwise of v.

Set $f_1 = r_F(v)$, $f_2 = r_F(f_1)$, ..., $f_h = l_F(u) = r_F(f_{h-1})$. Let $G' = \{g_1, ..., g_l\}$ be the subset of $G - F$ extreme in $F \cup G$, where the index is given by the ordering along $\partial U[u,v]$. We want to show that $r_{F\cup G}$ maps $f_1 \mapsto f_2 \mapsto ... \mapsto f_h \mapsto g_1 \mapsto ... \mapsto g_l \mapsto f_1$. We have already shown that $l_{F\cup G}$ maps f_1 to g_l, hence $r_{F\cup G}(g_l) = f_1$. By a similar argument $r_{F\cup G}(f_h) = g_1$. It is easy to see that $r_{F\cup G}(g_i)$ can only be g_{i+1}, for $1 < i < l$.

It remains to show that $r_{F\cup G}(f_i) = f_{i+1}$, for $1 < i < h$. This follows if we establish $span(f_i, F \cup G) = span(f_i, F)$ (which also establishes f_i extreme in $F \cup G$)). Since

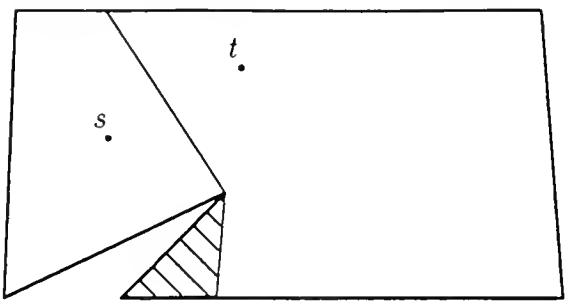


Figure 2.9: A configuration not in general position. The entire shaded region is contained in $b(s,t)$.

$f_h = l_F(u)$ is extreme in $F \cup G$, f_h is extreme in $F \cup \{v\}$, hence f_h is on the far side of F from v, and hence $f_1, ..., f_h$ are on the far side of F from v. Similarly $f_1, ..., f_h$ are on the far side of F from u. Since r and l preserve order on ∂U, the far side of F from any $w \in \partial U[u,v]$ includes $f_1, ..., f_h$. In particular, for any $w \in \partial U[u,v]$ and any i, $1 < i < h$, we have $f_i \notin g(l(w), r(w))$ (since f_i extreme in F and $f_i \neq l(w), r(w)$), and (as in the proof of Lemma 2.4.1), $\theta(f_i, w) \in \theta(f_i, g(l(w), r(w))) \subseteq \theta(f_i, R(F)) = span(f_i, F)$. Hence $span(f_i, F \cup G) = span(f_i, F)$.

Now if $g_1 = u$ and $g_l = v$, we are done. If, say, $g_1 \neq u$, then u must appear on $g(f_k, g_1)$. To obtain the Lemma split $g(f_k, g_1) = g(l_F(u), g_1)$ into $g(l_F(u), u)$ and $g(u, g_1)$, then merge $g(u, g_1)$ with $\partial U[g_1, g_k]$. A similar split applies if $g_l \neq v$. □

2.5 The General Position Assumption

For the remainder of our discussion of problems in the geodesic metric, we make the following *general position assumption* concerning the set of sites S in U: no reflex corner of ∂U is equidistant from two sites. This condition can always be satisfied by applying a slight perturbation to the positions of the sites or corners. If this assumption is not made, the set of points equidistant from two sites may include a two-dimensional region (cf. Figure 2.9). This Section contains some consequences

of the general position assumption that are critical to the rest of our analysis. We emphasize that none of Lemmas 2.5.1, 2.5.2, and 2.5.4 hold if the assumption is violated.

Lemma 2.5.1 *If s, t are distinct sites, $x \in U$, and $d_s(x) = d_t(x)$, then $\theta(x,s) \neq \theta(x,t)$.*

Proof: Suppose $\theta(x,s) = \theta(x,t)$. Then the geodesics $g(s,x)$ and $g(t,x)$ share their final link yx. Point y must be the anchor of x with respect to both of s and t. Now $y \neq s, t$ since otherwise $d(s,y) = d(t,y)$ would imply $s = t$. Hence y must be a reflex corner of ∂U equidistant from s and t, contradicting the general position assumption. □

Lemma 2.5.2 *Suppose $u, v \in U$, $u \neq v$, and s and t are distinct sites. If $d_s(u) \leq d_t(u)$ and $d_t(v) \leq d_s(v)$, $g(s,u)$ cannot intersect $g(t,v)$.*

Proof: Suppose x lies on both $g(s,u)$ and $g(t,v)$. Without loss of generality, assume $d(x,s) \leq d(x,t)$. Observe that

$$d_v(t) \leq d_v(s) \leq d_v(x) + d(x,s) \leq d_v(x) + d(x,t) = d_v(t).$$

Hence $d(x,s) = d(x,t)$, implying that the path $g(s,x) \cup g(x,v)$ is in fact geodesic $g(s,v)$, and similarly $g(t,x) \cup g(x,u)$ is $g(t,u)$. Since x lies on both $g(s,u)$ and $g(s,v)$, x must be a reflex corner of ∂U or an interior point of a common link of the two geodesics connecting two reflex corners of ∂U. In the second case the link must be common to $g(t,u)$ and $g(t,v)$ as well, hence in either case there is a reflex corner of ∂U equidistant from s and t, violating the general position assumption. □

Corollary 2.5.3 *Suppose $u, v \in U$, $u \neq v$, and each of u and v is equidistant from two distinct sites s and t. Then $g(s,u)$ does not intersect $g(t,v)$.*

Lemma 2.5.4 *There is at most one point equidistant from three distinct sites.*

Proof: Suppose to the contrary that points u and v are both equidistant from sites r, s, t. First note that r, s, t cannot lie on a common geodesic, for if say $r \in g(s,t)$ then by Lemma 2.2.1, $d_u(r) < \max(d_u(s), d_u(t))$. Hence r, s, t are extreme points of $\{r, s, t\}$.

We claim that r, s, t are extreme points of $\{r, s, t, u\}$ (and also of $\{r, s, t, v\}$). In order to demonstrate this, we show $r \notin R(\{s, t, u\})$. Now r does not lie on $g(u, s)$ or $g(u, t)$, else s or t would be further from u than r. As argued before, r does not lie on $g(s, t)$. If r is in the interior of $R(\{s, t, u\})$ then by the Triangle Lemma, $g(u, \bar{r})$ intersects $g(s, t)$, where $\bar{r}$ is a shadow of $g(u, r)$. But then using Lemma 2.2.1 again, we would have $d_u(r) < \max(d_u(s), d_u(t))$.

Now suppose one of u and v, say u, is extreme in $R(\{r, s, t, u, v\})$. Hence r, s, t, u are extreme in $R(\{r, s, t, u\})$; assume that they appear in that counterclockwise order. By Lemma 2.3.6, $r \in U[u, s]$, $t \in U[s, u]$, and $r, t \notin \bar{g}(u, s)$. It must be that $\bar{g}(u, s) = g(u^*, \bar{s})$ intersects either $g(v, r)$ or $g(v, t)$, assume it is $g(v, t)$. To obtain a contradiction of Corollary 2.5.3, we show that in fact $g(u, s)$ intersects $g(v, t)$. Now uu^* intersects $R(\{r, s, t, u, v\})$ only at u since u is extreme in $R(\{r, s, t, u, v\})$. Hence uu^* is disjoint from $R(\{r, s, t, v\})$. Also $s\bar{s}$ intersects $R(\{r, s, t, v\})$ only at s, since s is extreme in $R(\{r, s, t, v\})$ and some portion of $g(u, s)$ must lie in $R(\{r, s, t\})$, hence in $R(\{r, s, t, v\})$. Since $g(v, t) \subseteq R(\{r, s, t, v\})$, $g(v, t)$ must intersects $g(u, s)$.

If neither u nor v is extreme in $R(\{r, s, t, u, v\})$ then both $u, v \in R(\{r, s, t\})$ and the proof is similar. Geodesics $g(u, r)$, $g(u, s)$, and $g(u, t)$ split $R(\{r, s, t\})$ into three geodesic triangles. Hence v lies in one of the triangles which again implies a contradiction of Corollary 2.5.3. □

Remark: Unlike standard Euclidean geometry, where a point equidistant from three sites always exists (unless the sites are collinear), this need not be the case with respect to the geodesic distance, intuitively because the desired point may have to lie outside of the universe U. For example, even if U is convex, so that the Euclidean and the geodesic metrics coincide, the point equidistant from three nearly-collinear sites lies arbitrarily far from them and in particular, may land outside of U.

2.6 Proximity Considerations

Having explored the immediate implications of the general position assumption, we finally come to consider some crucial *metric* properties of geodesic geometry in a simple polygon. Namely, we introduce the notion of a bisector and study the (almost trivial) closest-site and furthest-site geodesic Voronoi diagrams of *two* sites in U; they correspond to a natural partition of U induced by the distance functions from these two sites. The next two Chapters are concerned exclusively with the structure and complexity of the two Voronoi diagrams of a general set of point sites and efficient procedures for their computation.

The *bisector* $b(s,t)$ of two distinct sites s and t is $\{x \in U \mid d_s(x) = d_t(x)\}$ and the *half space closer to s than t*, $H(s,t)$, is $\{x \in U \mid d_s(x) < d_t(x)\}$. Clearly $H(s,t)$, $b(s,t)$, and $H(t,s)$ form a partition of U. A *breakpoint* of $b(s,t)$ is the intersection of $b(s,t)$ with a shortest path partition edge from s or t. Figure 2.10 indicates the bisectors of three points, with breakpoints marked. Notice that we define two *distinct* notions corresponding to Euclidean half plane–one to represent the portion of U cut off by a "straight line" (cf. Section 2.1) and one to reflect the natural partition of U induced by the distances to two distinct sites. The remainder of this Section is devoted to the properties of the bisectors and half spaces under the assumption of general position. We show that a bisector is a smooth connected curve that stays away from ∂U except for its endpoints and a half space is star-shaped (with respect to the site it contains) and therefore, connected and simply-connected. Note, however, that a half space is in general not relatively convex.

A set $Q \subseteq U$ is *star-shaped around* $x \in U$ (with respect to the geodesic metric) if $g(x,y) \subset Q$ whenever $y \in Q$. By an argument identical to that of Lemma 2.3.1, a star-shaped subset of U is necessarily simply connected.

Lemma 2.6.1 *For any two sites s and t in general position and any $x \in b(s,t)$:*

1. $g(s,x) - \{x\} \subseteq H(s,t)$,

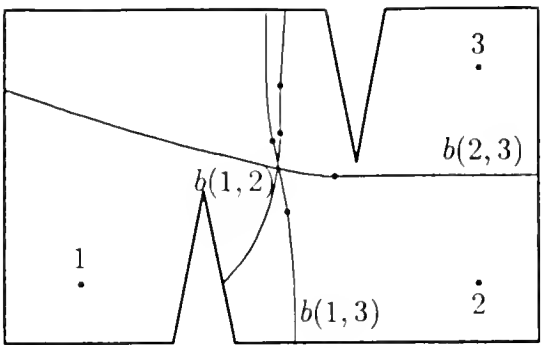


Figure 2.10: Bisectors; dots indicate breakpoints.

2. *for any extension* $g(s,y)$ *of the geodesic* $g(s,x)$, $g(x,y) - \{x\} \subseteq H(t,s)$, *and*

3. *for any point* $y \in U$, $g(s,y)$ *intersects* $b(s,t)$ *in at most one point.*

Proof: Once again, notice that the Lemma is false in absence of the general position assumption (cf. Figure 2.9).

(1) Suppose there exists a point $y \in g(s,x) - \{x\}$ that does not lie in $H(s,t)$. Then $d_s(y) \geq d_t(y)$ while $d_s(x) = d_t(x)$, which implies $g(y,t) \cap g(x,s) = \emptyset$ by Lemma 2.5.2—a contradiction.

(2) Suppose there exists an extension $g(s,y)$ of $g(s,x)$ so that $y \neq x$ and $y \notin H(t,s)$. Then $d_s(y) \leq d_t(y)$, contradicting Lemma 2.5.2, as $d_s(x) = d_t(x)$ and $x \in g(s,y) \cap g(x,t)$.

(3) Immediate from (1). □

Corollary 2.6.2 *Both* $H(s,t)$ *and* $H(s,t) \cup b(s,t)$ *are star-shaped around* s.

Corollary 2.6.3 *For any* $x \in b(s,t) - \partial U$, *there are points of both* $H(s,t)$ *and* $H(t,s)$ *arbitrarily close to* x. *Thus* $b(s,t)$ *is the relative boundary of both* $H(s,t)$ *and* $H(t,s)$.

Lemma 2.6.4 *Bisector* $b(s,t)$ *is a smooth curve connecting two points on* ∂U *and having no other points in common with* ∂U. *It is the concatenation of line segments and hyperbolic arcs; the points where such arcs meet are precisely the breakpoints of* $b(s,t)$. *The tangent to* $b(s,t)$ *at* x *bisects the angle between* $\theta(x,s)$ *and* $\theta(x,t)$.

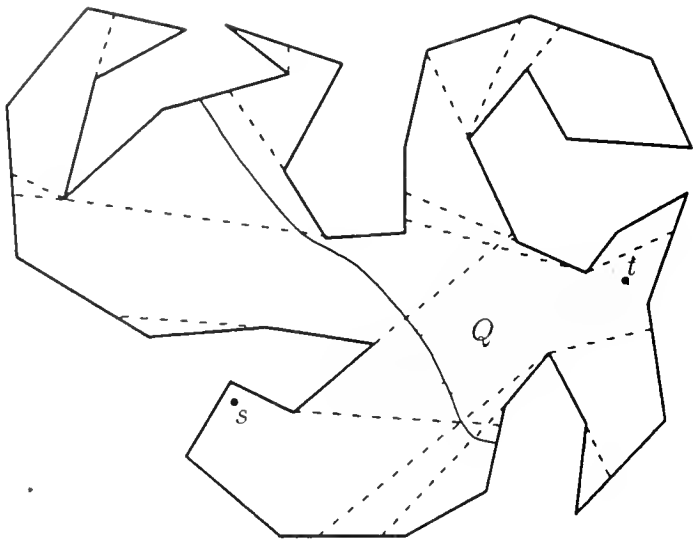


Figure 2.11: An illustration to the proof of Lemma 2.6.4

Proof: Refer to Figure 2.11. First of all, observe that $H(s,t)$ is connected to ∂U since, for any foreshadow s^* of $g(s,t)$, $d_s(s^*) < d_t(s^*)$. Similarly, $H(t,s)$ is connected to ∂U.

Partition U into polygonal regions by removing from it $\bigcup_{a \in V} \{p_a(s), p_a(t)\}$. Let Q be the closure of one of the resulting regions. By construction, there are two anchors $\hat{s}$ and $\hat{t}$ such that, for any point x in Q, $d(s,x) = d(s,\hat{s}) + |\hat{s} - x|$ and $d(t,x) = d(t,\hat{t}) + |\hat{t} - x|$, where points are regarded as vectors and $|...|$ denotes the Euclidean norm. Thus the condition that x lie on $b(s,t)$ is equivalent in Q to $d(s,\hat{s}) + |\hat{s} - x| = d(t,\hat{t}) + |\hat{t} - x|$. Notice that, if $b(s,t)$ meets Q, $\hat{s} \neq \hat{t}$, as otherwise the reflex vertex $\hat{s} = \hat{t}$ would be equidistant from s and t, violating the general position assumption. Therefore, $b(s,t) \cap Q$ consists of zero or more arcs of a hyperbola Γ with foci $\hat{s}$ and $\hat{t}$. Γ is a straight line if $d(s,\hat{s}) = d(t,\hat{t})$. By elementary analytic geometry, the tangent to Γ bisects the angle between the directions to $\hat{s}$ and to $\hat{t}$. (It is also possible for $\Gamma \cap Q$ to contain isolated points.)

Hence $b(s,t)$ is the union of (relatively closed) straight and hyperbolic arcs and

discrete points. It can be thought of as an embedded planar graph whose removal from U breaks U into exactly two connected components, namely, $H(s,t)$ and $H(t,s)$. First of all, observe that $b(s,t)$ cannot contain cycles, for otherwise one of the two half spaces will be separated from ∂U. Note that a cycle touching ∂U is excluded as well, as otherwise the enclosed half-space would not contain a single point of ∂U. Second, no component of $b(s,t)$ can be confined to the interior of U, as that would mean that one of the half spaces is not simply connected. In addition, by Corollary 2.6.3, $b(s,t)$ cannot have degree-1 vertices in the interior of U, meet ∂U in more than a discrete set of points, or have isolated vertices, for otherwise there would exist points on $b(s,t)$ that are not limit points of both $H(s,t)$ and $H(t,s)$. Thus $b(s,t)$ is a forest with all leaves on ∂U. In particular, it must be a single path, as its removal breaks U into two connected components.

Finally, the directions $\theta(x,s)$ and $\theta(x,t)$ vary continuously along $b(s,t)$, so the angle bisection property of the individual hyperbolic arcs also holds at their joining points, implying that the individual arcs are "glued together" smoothly. □

Corollary 2.6.5 *Given $x \in b(s,t)$, $\theta(x,s)$ and $-\theta(x,t)$ enter $H(s,t)$ at x, while $-\theta(x,s)$ and $\theta(x,t)$ enter $H(t,s)$ (if they stay within U).*

Note: The fact that the bisector of two sites is a smooth curve whose tangent bisects the angle between the directions to the two sites is quite general. Not only does it (obviously) hold for point sites in the Euclidean metric, but it continues to hold for arbitrary convex sites in the Euclidean plane[LS87a].

Corollary 2.6.6 *The intersection $b(s,t) \cap p_a(s)$ consists of at most one point. If they do intersect, say, at point x, $p_a(s)$ cannot be tangent to $b(s,t)$ at x.*

Proof: The first claim is immediate from Lemma 2.6.1(3). By Lemma 2.6.4, $p_a(s)$ being tangent to $b(s,t)$ at x would force $\theta(x,s) = \theta(x,t)$, contradicting Lemma 2.5.1. □

Chapter 3

Closest-Site Geodesic Voronoi Diagram

In this Chapter we present an algorithm which calculates the closest-site Voronoi diagram of a set S of k point sites in a simple polygon U with n vertices, defined in terms of the geodesic metric. Our algorithm runs in time $O((n+k)\log(n+k)\log n)$, which is within a logarithmic factor of optimal and an order of magnitude faster than a previous algorithm of [AA87] (see the discussion below; the lower bound is argued in Section 4.3).

A notable property of the geodesic metric is that calculation of the shortest path between two points is not an easy operation and, without preprocessing, must take $\Omega(n)$ time in the worst case. Also, a geodesic bisector of two sites is, in the worst case, the concatenation of $\Theta(n)$ distinct straight and hyperbolic arcs, which may at first sight suggest that the worst-case overall complexity of the geodesic Voronoi diagram is $\Theta(nk)$. Fortunately, this is not the case, and we show that the total size of the diagram is only $\Theta(n+k)$, where only $O(k)$ of the diagram vertices are "true" (degree 3) vertices. Roughly speaking, the diagram can be refined by subdividing the cell of each site $s \in S$ according to the shortest path partition from s (cf. Section 2.2); the refined diagram can then be regarded as the union of all these shortest path partition edges, truncated to their corresponding Voronoi cells and separated from each other

by Voronoi edges.

The best previously known algorithm for constructing the closest-site geodesic Voronoi diagram of k point sites inside a simple polygon with n corners is due to Asano and Asano [AA87]; it runs in time $O(nk + n \log\log n + k \log k)$. The feature of their algorithm that necessitates quadratic time is the explicit construction of the shortest path tree for the *full* polygon from *each* site. As these trees are easily seen to have a total of nk distinct edges in the trivial case of a convex polygon, the (worst-case) quadratic bound on the running time of the algorithm follows. We have circumvented this problem by never building the *full* shortest path tree of the polygon for each of the sites, instead the tree (actually, the closely related shortest path partition of U) from each site is constructed only for the part of the polygon that can conceivably lie in the Voronoi cell of the site.

Our algorithm uses a familiar divide-and-conquer strategy for obtaining the closest-site geodesic Voronoi diagram. However, peculiarities of the geodesic metric necessitate a somewhat non-standard implementation of this strategy. For example, to avoid constructing the shortest path trees from every site, the relatively standard merge step must be preceded by a step that extends a recursively computed diagram of the subset of sites inside half the polygon to the full polygon. The extension phase is the least conventional part of our algorithm and requires sweeping the triangulation of the polygon by a polygonal scan-line.

Section 3.4 describes a simpler algorithm that can be used to compute the geodesic Voronoi diagram in time $O((n+k)\log(n+k))$, if the set S of sites contains all reflex vertices of U. In this case, Voronoi cells are (Euclidean) star-shaped and we are able to exploit this property to speed up the calculation of the diagram.

Possible applications of our algorithm include the closest pair problem, the post office problem and all-nearest-neighbors problem in the context of a polygonal universe, such as a (polygonal) island with no interior lakes or a polygonal factory floor. It is likely that other planar point location and proximity problems whose solutions employ Euclidean Voronoi diagram could be generalized to questions about internal

metric in a simple polygon and could take advantage of our algorithm. For example, one might wish to investigate how the analogues of a "Delaunay triangulation" and "minimum spanning tree" behave in the context of the geodesic metric.[1]

We start this Chapter by presenting the definition of the closest-site geodesic Voronoi diagram and some of its more important geometric properties (Section 3.1) and proceed to a detailed analysis of its fine structure and complexity (Section 3.2). We prove that the complexity of the diagram is $O(n + k)$ in the sense that it can be described as a union of that many openly disjoint objects, each of complexity bounded by a constant. Section 3.3 gives a detailed description of our algorithm for constructing the diagram. Section 3.4 outlines a simplified version of the algorithm for producing the closest-site geodesic Voronoi diagram for a set of sites that includes all reflex corners of the polygon. Section 4.3 mentions some related open problems.

3.1 Definitions

Recall that the *universe* U is a compact region in the plane whose boundary ∂U is a simple n-gon. Let V denote the set of its corners, and let $S \subseteq U$ be a set of k point sites. The *(geodesic closest-site) Voronoi cell* of site s is $V_U(s) = \bigcap_{t \neq s} H(s,t)$. The *(geodesic closest-site) Voronoi diagram* $\mathcal{V} = \mathcal{V}_U(S)$ is

$$\{x \in b(s,t) \mid s,t \in S \text{ and } d_s(x) = \min_{r \in S} d_r(x)\}.$$

Whenever it does not cause ambiguities, we will omit U and S in the above notation. Figure 3.1 shows the Voronoi diagram and Voronoi cells of three points. Notice that, if U is convex, $\mathcal{V}$ coincides with the standard Euclidean Voronoi diagram truncated to within U. A *Voronoi edge* $e(s,t)$ is

$$e(s,t) = \mathcal{V} \cap b(s,t) = b(s,t) \cap \bigcap_{r \neq s,t} (H(s,r) \cup b(s,r)).$$

[1] One generalization of Delaunay triangulation (briefly discussed in Section 3.4) was investigated by Lee and Lin [LL86], who defined it directly, and not as the dual of the Voronoi diagram.

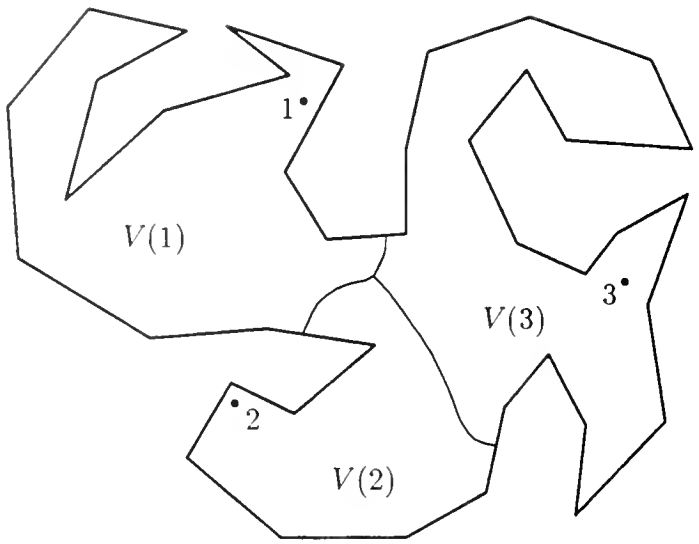


Figure 3.1: Closest-site geodesic Voronoi diagram.

If the intersection consists of more than one point (else we will say that $e(s,t)$ does not exist). Notice that, under the general position assumption, as edge is indeed a one-dimensional object. A *(Voronoi) vertex* is a point $x \in \mathcal{V}$ which has three or more sites simultaneously closest to it. By Lemma 2.5.4, there is at most one such point x for each triple of sites. A *hitpoint* is the intersection of a Voronoi edge with ∂U. Intuitively, a hitpoint corresponds to the "point at infinity" of an infinite Voronoi edge in a Euclidean closest-site Voronoi diagram.

Lemma 3.1.1 *Each Voronoi edge $e(s,t)$ is connected and has vertices or hitpoints as endpoints.*

Proof: Suppose r, s, t are distinct sites. Since d_r is continuous, a connected component of $b(s,t) \cap (H(s,r) \cup b(s,r))$ must have for each of its endpoints either an endpoint of $b(s,t)$, i.e., a hitpoint, or a point equidistant from s, t, r. But by Lemma 2.5.4, there is at most one point equidistant from s, t, r, so $b(s,t) \cap (H(s,r) \cup b(s,r))$ consists of a single connected component. Hence also $e(s,t) = b(s,t) \cap \bigcap_{r \neq s,t} (H(s,r) \cup b(s,r))$ is connected and has hitpoints or vertices for endpoints. □

Lemma 3.1.2 *Suppose that $s_1, ..., s_k, s_{k+1} = s_1$ are the sites closest (and thus equidistant) from vertex v, and that directions $\theta(v, s_1), ..., \theta(v, s_k)$ are in counterclockwise order. Then, for each i, edge $e(s_i, s_{i+1})$ is incident to v and extends away from v in direction bisecting $\angle s_i v s_{i+1}$, if that direction (locally) stays inside U.*

Proof: Elementary analysis, using $\nabla_v d_{s_i}(v) = -\theta(v, s_i)$. □

The objective of our algorithm is to obtain the decomposition of U into Voronoi cells or, equivalently, compute $\mathcal{V}$ as a planar embedded graph. Observe that applying known techniques for planar point location to such a planar map allows efficient computation of the set of all sites closest to an arbitrary query point $x \in U$.

Lemma 3.1.3 *Both $V(s)$ and $\mathrm{Cl}(V(s)) = \bigcap_{t \neq s} H(s,t) \cup b(s,t)$ are star-shaped around s, where* Cl *denotes the closure of a set.*

Proof: Star-shapedness is immediate from Corollary 2.6.2. The form of $\mathrm{Cl}(V(s))$ follows from Lemma 2.6.1. □

For a proof of the fact that, for a rather large class of distance measures, the Voronoi cell $V(s)$ of a (point) site s is always star-shaped around s with respect to the distance used to define $V(s)$, see, for example, [Aur88].

3.2 Fine Structure of the Diagram

Since Voronoi cells are connected, $\mathcal{V} \cup \partial U$ is a planar map with k bounded faces and all vertices of degree at least three. Hence, by Euler's formula, the number of Voronoi edges and vertices, as well as hitpoints, is only $O(k)$. However, this estimate does not faithfully reflect the (size) complexity of $\mathcal{V}$, as a Voronoi edge may consist of as many as $\Theta(n)$ hyperbolic and straight arcs, joined at breakpoints.

Let us estimate the total number of breakpoints on Voronoi edges. We claim that the number of such points is $O(n)$, thus bounding the total complexity of the map in question by $O(n+k)$, and proving that the size of the desired Voronoi diagram is linear

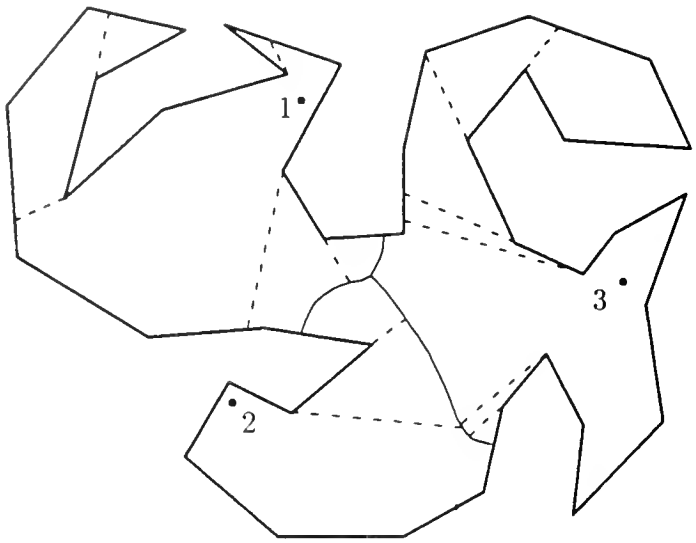


Figure 3.2: Refined Voronoi diagram of the sites of Figure 3.1.

in the size of the input. Roughly speaking, we augment the planar map induced by $\mathcal{V}$ by adding to it the shortest path partition segments $p_a(s)$ truncated to $V(s)$, for each $s \in S$. Segments $p_a(s)$ that do not intersect $V(s)$ are simply discarded. Note that segment $p_a(s)$ is *not* discarded if and only if $a \in V(s)$ (Figure 3.2). More formally, we define a refinement of the Voronoi partition, obtained by further subdividing each Voronoi cell $V(s)$ by the shortest path partition from s. Each bounding edge of a refined Voronoi cell is a line segment or a section of a single hyperbolic arc. The main consequence of this construction is a linear bound on the complexity of the refined Voronoi diagram. This implies an $O(n + k)$ bound on the complexity of the Voronoi diagram itself.

The *refined Voronoi cell of site s with anchor a*, $V_a^*(s)$, is $V(s) \cap P_a(s)$. The *refined bisector edge* $e_{ab}^*(s,t)$ is $e(s,t) \cap P_a(s) \cap P_b(t)$. The *refined partition edge (from s with anchor a)*, $p_a^*(s)$, is $V(s) \cap p_a(s)$. A *refined Voronoi edge*is a refined bisector edge or a refined partition edge. Observe that distinct refined Voronoi edges are disjoint (except possibly at their endpoints). Empty refined Voronoi cells and edges

and refined Voronoi edges consisting of a single point are disregarded.

Suppose $e^*_{ab}(s,t)$ is not empty. It is easy to see that each endpoint of $e^*_{ab}(s,t)$ is either a vertex of $\mathcal{V}$, a hitpoint, or a breakpoint. Moreover, $e^*_{ab}(s,t)$ does not contain breakpoints (except possibly as endpoints) and is connected, as $b(s,t)$ visits each $P_a(s)$ at most once (by Lemma 2.6.4 and Corollary 2.6.6). Consequently $e^*_{ab}(s,t)$ is a hyperbolic arc or a line segment.

Lemma 3.2.1 *Either $p^*_a(s)$ is empty, or it is all of $p_a(s)$, or it has an open endpoint at a breakpoint of $e(s,t)$ for some site t and closed endpoint at a. (The latter two cases are illustrated in Figure 3.2.)*

Proof: Suppose $p^*_a(s)$ is not empty and is not all of $p_a(s) = ay$ (where $y \in \partial U$). Then $p_a(s)$ must intersect some edge $e(s,t)$ the first time it leaves $V(s)$. By Lemma 2.6.1, the intersection is a single point x, and by star-shapedness of $V(s)$ and $\mathrm{Cl}(V(s))$, $p^*_a(s) = ax - \{x\}$ is contained in $V(s)$. □

The *refined (closest-site geodesic) Voronoi diagram* $\mathcal{V}^*_U(S)$ is the union of the refined Voronoi edges.

Recall that the breakpoints of bisector $b(s,t)$ are the points where it meets either a segment $p_a(s)$, for some $a \in H(s,t)$ or a segment $p_b(t)$, for some $b \in H(t,s)$. An easy induction implies that breakpoints of $e(s,t)$ occur precisely at the points where it meets $p_a(s)$ for some $a \in V(s)$ or $p_b(t)$ for some $b \in V(t)$. In the former case $p^*_a(s) = V(s) \cap p_a(s)$ is neither empty nor all of $p_a(s)$; the latter case is symmetric. Hence each breakpoint of $e(s,t)$ is incident with (at least) one refined partition edge. However, any corner a lies in exactly one Voronoi cell and has at most one refined partition edge emanating from it (namely, $p^*_a(s)$). Hence there are no more than $O(n)$ breakpoints.

Theorem 3.2.2 *The complexity of $\mathcal{V}^*_U(S)$ (and thus of $\mathcal{V}_U(S)$) is $O(n+k)$ where n is the number of corners of U and $k = |S|$.*

3.3 The Algorithm

We begin this Section with an overview of our algorithm for computing the (refined) geodesic Voronoi diagram $\mathcal{V}^*$ of a set of sites S in a simple polygon U. The steps that require more detailed treatment are discussed at greater length in the following subsections.

Algorithm 3.1
Input: A compact region U bounded by a simple n-gon and a set $S \subset U$ of k point sites in general position.
Output: The refined closest-site geodesic Voronoi diagram $\mathcal{V}_U^*(S)$.

1. **Preprocessing**

   (i) Triangulate U in $O(n \log n)$ time [GJPT78].

   (ii) Identify the triangle containing each site in total $O((n+k)\log n)$ time by using an appropriate planar point location algorithm (e.g., that of [ST86]).

   (iii) In $O(n \log n)$ time compute a balanced decomposition of the triangulation tree [Cha82] so that U can be recursively cut by a chord into two parts each having at least one quarter of the number of sides, in constant time per cut (see also [GHL*87]).

2. **Recursive body**

   (i) If S consists of a single site s, $\mathcal{V}_U(S)$ is empty. There is a single cell $V_U(s) = U$. $\mathcal{V}_U^*(S) = \bigcup p_a(s)$, where a ranges over all reflex vertices of U, is then computed by utilizing the linear-time shortest path partition construction of [GHL*87]; otherwise

   (ii) If U is a triangle, $\mathcal{V} = \mathcal{V}^*$. Compute the Euclidean closest-site Voronoi diagram of S in $O(k \log k)$ time and truncate it to U in $O(n)$ time, recording the intersections of the Voronoi edges with ∂U; otherwise

(iii) Split U by a chord into two roughly equal polygons U_L and U_R and divide S into S_L and S_R with $S_L \subset U_L$ and $S_R \subset U_R$. This can be accomplished in constant time, as a balanced decomposition of U has been precomputed and the sites are already associated with the triangles containing them, so that partitioning S is implicit and requires no processing.

(iv) Recursively compute $\mathcal{V}^{\bullet}_{U_L}(S_L)$ and $\mathcal{V}^{\bullet}_{U_R}(S_R)$.

(v) Extend $\mathcal{V}^{\bullet}_{U_R}(S_R)$ to $\mathcal{V}^{\bullet}_{U}(S_R)$ and $\mathcal{V}^{\bullet}_{U_L}(S_L)$ to $\mathcal{V}^{\bullet}_{U}(S_L)$ in $O((n+k)\log(n+k))$ time as described in Section 3.3.1.

(vi) Compute $\mathcal{V}^{\bullet}_{U}(S)$ by merging $\mathcal{V}^{\bullet}_{U}(S_L)$ and $\mathcal{V}^{\bullet}_{U}(S_R)$ in $O(n+k)$ time. The details of this step can be found in Section 3.3.2.

In part 2 of the algorithm, if either S_L or S_R is empty, omit the corresponding recursive call and skip the merge phase.

As preprocessing takes $O((n+k)\log n)$ time and the recursive part is called on two subproblems of sizes $(\beta n, k_1)$ and $((1-\beta)n, k_2)$, respectively, where $k_1 + k_2 = k$ and $\frac{1}{4} \leq \alpha \leq \frac{3}{4}$, with the solution obtained from partial solutions, if any, in time $O((n+k)\log(n+k))$, a simple recurrence bounds the running time of the algorithm by $O((n+k)\log(n+k)\log n)$, as claimed. Observe that our algorithm divides U into roughly equal parts, while the standard divide-and-conquer approach to the computation of Voronoi diagrams splits the set of sites.

3.3.1 Extending the Diagram

In this Section we will describe a procedure implementing step 2(v) of Algorithm 3.1. It computes the refined closest-site geodesic Voronoi diagram for a set of sites in a simple polygon U given the diagram for the same sites in a subpolygon, where the subpolygon was obtained by cutting U along a chord, as in step 2(iii) of Algorithm 3.1.

Let U be a closed region bounded by a simple n-gon. Assume that a triangulation of U is available and let e be a chord of the triangulation. Denote (the closures

of) the two polygonal regions into which e cuts U by U_1 and U_2. Suppose S is a set of k sites in U_1. We will describe an algorithm for extending $\mathcal{V}^*_{U_1}(S)$ to U_2 and thereby obtaining $\mathcal{V}^* = \mathcal{V}^*_U(S)$. Extension of the refined Voronoi diagram proceeds by traversing the triangulation tree of U_2 (which is the planar graph whose vertices are triangles and in which two triangles are connected by an edge whenever they share an edge of the triangulation) in a top-down manner starting from the triangle adjacent to e. "Visiting" a triangle Δ involves extending the refined Voronoi diagram to Δ and preparing the data structures for further extension to unvisited triangles adjacent to Δ. Construction of the refined Voronoi diagram in the interior of Δ is performed by a sweeping algorithm. Before we describe the algorithmic details, some properties of $\mathcal{V}^*$ restricted to U_2 need to be examined. In particular, we will show that the edge structure of $\mathcal{V}^*_U(S)$ in the interior of U_2 is a forest constructible by sweeping.

Consider the Voronoi diagram $\mathcal{V} = \mathcal{V}_U(S)$. Recall that $\mathcal{V}$ and $\mathcal{V}^*$ are planar embedded graphs. In particular, we will define $\overline{\mathcal{V}} = \mathcal{V} \cap U_2 = \mathcal{V}_U(S) - \mathcal{V}_{U_1}(S)$ as the graph $\mathcal{V}$ restricted to U_2, i.e., $\mathcal{V}$ with vertices lying outside of U_2 deleted and edges leaving U_2 truncated down to their points of intersection with e. We define $\overline{\mathcal{V}^*} = \mathcal{V}^* \cap U_2 = \mathcal{V}^*_U(S) - \mathcal{V}^*_{U_1}(S)$ similarly. For an illustration, see Figure 3.3.

Lemma 3.3.1 *$\overline{\mathcal{V}}$ is a forest (i.e., a cycle-free undirected graph) with leaves on ∂U_2.*

Proof: Suppose it contains a non-trivial cycle. Since we identify $\mathcal{V}$ with its embedding in U, we may speak of a *minimal* simple cycle γ–one whose closure does not contain another cycle. In particular, the interior of γ contains a connected component of $U - \mathcal{V}$, i.e., a cell of the Voronoi diagram. Thus there is a Voronoi cell $V(s)$ entirely contained in the interior of U_2. However, by Lemma 3.1.3, $V(s)$ is a relatively open set containing s, contradicting the assumption $S \subset U_1$. □

Corollary 3.3.2 *$\overline{\mathcal{V}^*}$ is a forest with leaves on ∂U_2.*

Proof: Since, in terms of its embedding in U, $\mathcal{V}^*$ differs from $\mathcal{V}$ only by the refined partition edges, it is sufficient to show that such edges can never close a cycle in $\mathcal{V}$.

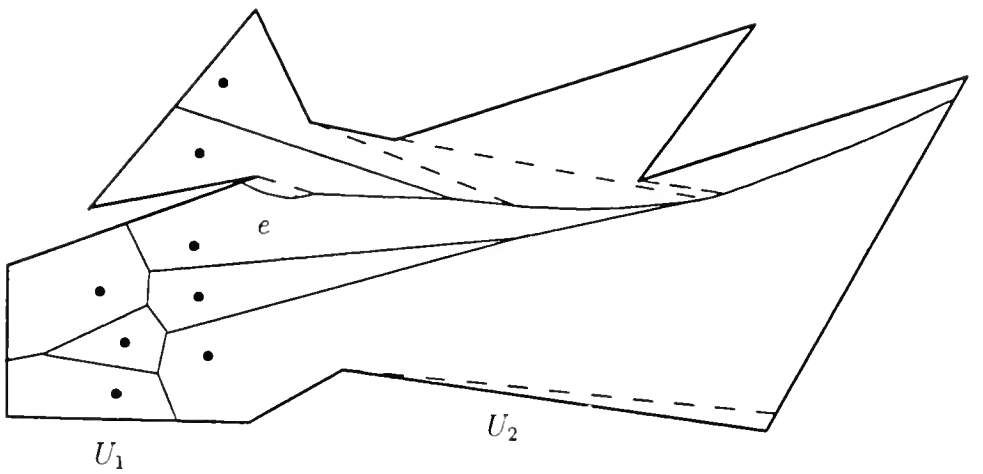


Figure 3.3: The structure of $\overline{\mathcal{V}^*}$.

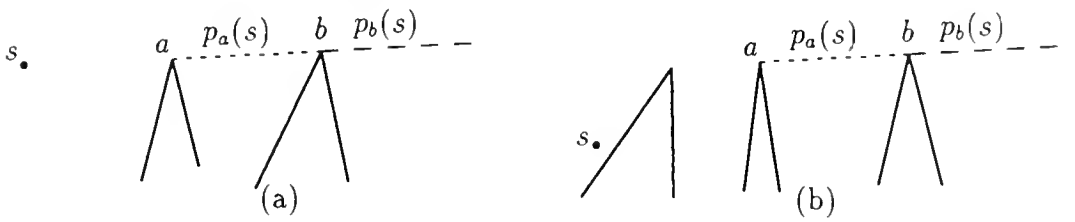


Figure 3.4: Degenerate refined partition segments. In both cases, $p_a(s)$ is degenerate.

It is clearly enough to establish that for any fixed site s, no connected component of $\bigcap_{a \in V(s)} \mathrm{Cl}(p_a^*(s))$ meets $\mathcal{V}$ twice. First of all, the closure of a refined partition edge $p_a(s)$ meets $\mathcal{V}$ in at most one point, namely, its endpoint furthest from s. Hence a single $p_a(s)$ cannot close a cycle in $\mathcal{V}$. As refined Voronoi edges are disjoint except for their endpoints, it is easily verified that the only way in which closures of two refined partition edges can intersect is for one of them to end at the anchor of the other (refer to Figure 3.4). In such a case, as there is at most one refined partition edge emanating from any reflex corner, any connected component of $\bigcap_{a \in V(s)} \mathrm{Cl}(p_a^*(s))$ is contained in a single geodesic passing through s, and thus still meets the relative boundary of $V(s)$ exactly once, as claimed. □

For convenience, if the non-anchor endpoint of a refined partition edge coincides with the anchor of another edge, we will think of the two endpoints as distinct. Note that this situation can only occur in "degenerate" configurations when either three vertices of U, or two vertices of U and a site of S are collinear. From now on we will think of the closures of refined partition edges as effectively pairwise disjoint.

We now proceed to analyze the structure of the forest $\overline{\mathcal{V}^*}$ in order to justify the sweeping algorithm for computing it, which is given at the end of this subsection. To clarify the terminology used in the following Lemmas, let us describe the manner in which the extension algorithm traverses the triangulation of U_2. It starts by visiting the triangle of U_2 adjacent to the edge e separating U_1 from U_2. In general, having entered triangle Δ through one of its edges, the algorithm marks the (at most) two unvisited triangles adjacent to Δ for later traversal. The algorithm advances by choosing an arbitrary marked triangle and visiting it. "Visiting" Δ involves constructing $\overline{\mathcal{V}^*}$ in Δ and preparing the data structures necessary for visiting the neighbors of Δ. The advance of the algorithm can be compared to gradual "flooding" of U_2, triangle by triangle.

Lemma 3.3.3 *Consider a triangle Δ which was entered through its side f in preorder traversal of the triangulation tree of U_2 from (the triangle incident to) e. Let x be a point in the interior of Δ on an edge σ of $\overline{\mathcal{V}^*}$. Then the line p tangent to σ at x crosses f.*

Proof: The Lemma is trivially true if $\sigma = p_a^*(s)$ as σ follows the shortest path from s and thus must enter Δ through f–there can be no anchors in the interior of Δ.

Consider an edge $\sigma = e_{ab}(s,t)$ (see Figure 3.5). Since $g(s,x)$ and $g(t,x)$ enter Δ through f and there are no anchors in the interior of Δ, the last links of both geodesics cut f. But by Lemma 2.6.4 p bisects the angle between $\theta(x,s)$ and $\theta(x,t)$, so it must also meet f. □

Lemma 3.3.4 *Let Δ and f be as above. Then all edges of $\overline{\mathcal{V}}$ that meet f intersect it transversally.*

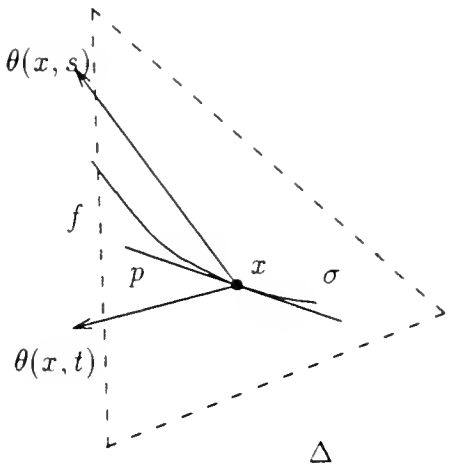


Figure 3.5: Illustration to the proof of Lemma 3.3.3.

Proof: Let f' and f'' be the two remaining sides of Δ. Assume $e(s,t)$ is tangent to f at x. Note that x is an interior point of f by the general position assumption. Since neither $g(s,x)$ nor $g(t,x)$ meet the interior of Δ, both $\theta(x,s)$ and $\theta(x,t)$ must lie in the closed half-plane bounded by f and not containing Δ. The only possible way of keeping this consistent with Lemma 2.6.4 that states that the tangent to $e(s,t)$ at x, i.e., f, must bisect the angle between $\theta(x,s)$ and $\theta(x,t)$ is for the latter two vectors to coincide and point along f, contradicting Lemma 2.5.1. □

Notice that the obvious extension of the previous Lemma to all edges of $\overline{\mathcal{V}^*}$ does not necessarily hold, since a refined partition edge may overlap f. Hereafter we will refer to such refined partition edges as *degenerate*.

We now proceed to define a "natural" orientation on $\overline{\mathcal{V}^*}$ (intuitively, "away from e,") that transforms $\overline{\mathcal{V}^*}$ into a root-directed forest. Orient edges of $\overline{\mathcal{V}^*}$ as follows: Again, let Δ be a triangle of U_2 entered through edge f. Given an edge σ, a point x on σ in the interior of Δ or on f (but not coinciding with either endpoint of f), the tangent p to σ at x intersects f transversally. Consider the traversal of σ that moves (locally at x) *away* from $f \cap p$ (Figure 3.6a). In case $x \in f$ direct σ *into*

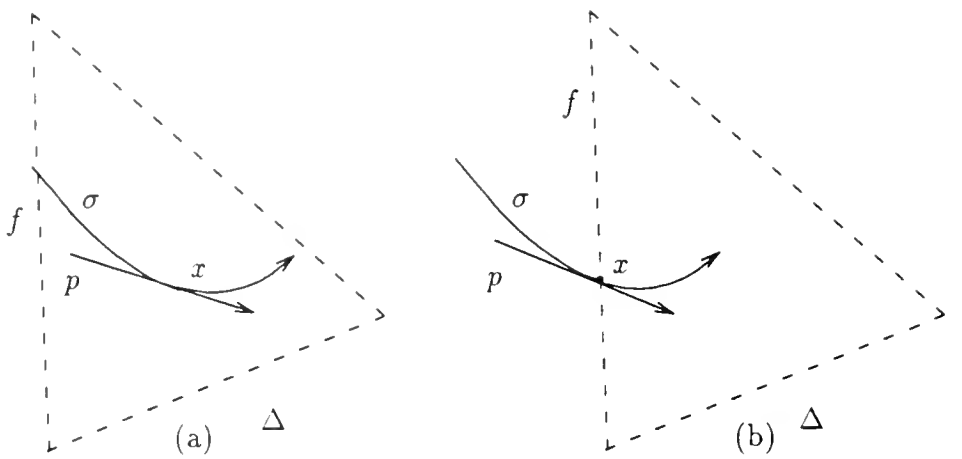


Figure 3.6: Orienting the edges of $\overline{\mathcal{V}^*}$.

Δ (Figure 3.6b). A degenerate refined partition edge that overlaps f is oriented so as to emanate *from* its anchor. We claim that this orientation is well-defined and indeed makes $\overline{\mathcal{V}^*}$ a root-directed forest. It is sufficient to demonstrate that the above convention assigns an orientation to each edge independent of the choice of point x and that no vertex has out-degree greater than one.

Lemma 3.3.5 *The orientation of any edge σ of $\overline{\mathcal{V}^*}$ is well-defined, i.e., there is a traversal of σ consistent with the above requirements at every point of σ.*

Proof: It is clear that the orientation of a degenerate refined partition edge cannot be inconsistent as it is defined globally rather than locally. Consider any other type of edge σ. Let us start by restricting our attention to a triangle Δ entered through its side f. The orientation is defined in terms of where the tangent to σ intersects f. Whether σ is a refined partition edge or a refined Voronoi edge, it is smooth and, as point x moves along σ, the tangent to σ at x continues to intersect f *on the same side* of the point of tangency, so the orientation of σ is consistent on a single connected portion of σ in the interior of Δ. The ambiguous case of the tangent coinciding with f is excluded by Lemma 3.3.4. Thus it is sufficient to show that the orientation stays

consistent when crossing chords separating triangles. And indeed it does, for suppose σ crosses the edge $f' \neq f$ separating Δ from Δ' (so Δ' must be entered though f'). By Lemma 3.3.4, σ intersects f' transversally. By smoothness of bisectors (Lemma 2.6.4), the tangent to σ at $\sigma \cap f'$ still meets f. According to our definition of orientation, σ will be oriented outward from Δ in the interior of Δ near f', it will be oriented from Δ to Δ' at its intersection point with f' and it will be oriented inward Δ' in the interior of Δ' near f', thus making the orientation consistent everywhere along σ. Note that σ cannot reenter Δ, as that would require a change in the orientation of σ between two consecutive appearances in Δ. □

So the orientation of an edge is well-defined. Let us proceed to classify the vertices of the forest $\overline{\mathcal{V}^*}$ and to prove that the orientation indeed yields a root-directed forest. We observe that the leaves of the directed forest, i.e., vertices with zero in-degree are exactly the points of intersection of $\mathcal{V}^*$ and ϵ together with reflex corners of ∂U_2 that are anchors of refined partition edges of $\mathcal{V}^*$. These vertices of $\overline{\mathcal{V}^*}$ indeed have degree one and their incident edges are directed away from them. An internal vertex of $\overline{\mathcal{V}^*}$ (i.e., neither a root nor a leaf) in the interior of Δ has out-degree of exactly one, as shown by the following Lemma. The case of an internal vertex landing on a chord of the triangulation will be handled separately (see the note following Lemma 3.3.7).

Lemma 3.3.6 *Every vertex v of $\overline{\mathcal{V}^*}$ in the interior of Δ has out-degree one.*

Proof: Suppose there are two or more edges emanating from v. Note that they are refined bisector edges and thus v is necessarily a Voronoi vertex, for refined partition edges do not *emanate* from a point in the interior of U. So suppose there are two or more refined bisector edges leaving v. Choose a pair of adjacent edges. By Lemma 3.1.2, the two edges must be portions of $e(s,t)$ and $e(t,r)$, for some sites s, t, r, and there must be a Voronoi cell $V_U(t)$ (locally) wedged between them (refer to Figure 3.7a). Note that, by definition of orientation, the tangents l_1 to $b(s,t)$ and l_2 to $b(t,r)$ at v intersect f when extended back through v. By construction, $V(t)$ is (locally) wedged between the half-lines of l_1 and l_2 extending *away* from f (see Figure 3.7a). In particular, by star-shapedness of $V(s)$ and $\mathrm{Cl}(V(s))$, $\theta(v,t)$ must

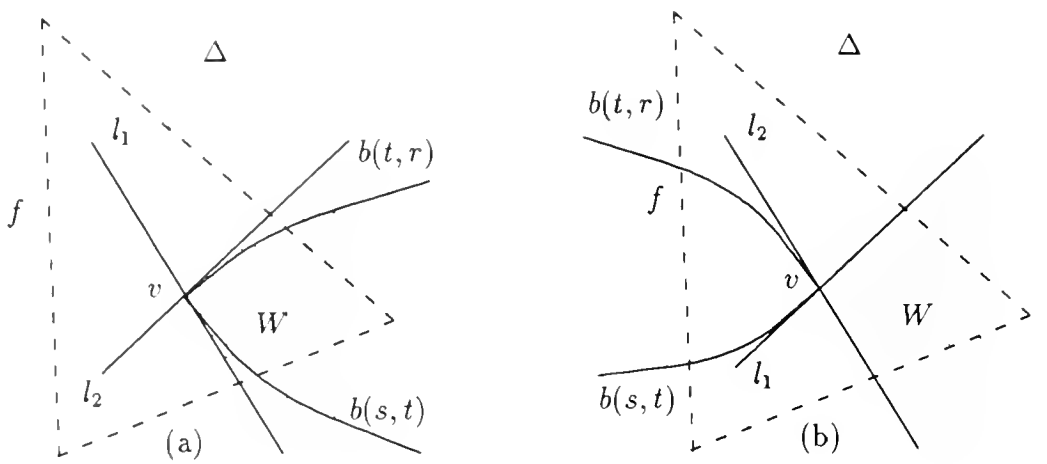


Figure 3.7: Internal vertices have out-degree 1.

point into the wedge W, contradicting the fact that t lies on f or on the side of f opposite Δ.

Suppose there is *no* edge emanating from v. If v is a breakpoint, two refined bisector edges join smoothly at v, thus guaranteeing the presence of at least one outgoing edge. Hence v must be a Voronoi vertex. As $\overline{V}$ has no leaves in the interior of Δ, there must be two or more Voronoi edges incident *to* v. Consider the tangents to these edges at v and choose the two edges whose tangents intersect f closest to either of its two endpoints (cf. Figure 3.7b). Similarly to the above argument, the larger wedge at v bounded by the two edges must (locally) belong to a single cell $V(t)$ for some $t \in S$, and the said Voronoi edges must be $e(s,t)$ and $e(t,r)$, respectively, for some $s, r \in S$. Let l_1 (l_2) be the line tangent to $b(s,t)$ (resp. $b(t,r)$) at v. Let W be the smaller wedge at v bounded by the half-lines of l_1 and l_2 that do not intersect f. By star-shapedness of $V(t)$, $\theta(v,t)$ must point into W, because it bisects the angle between l_1 and l_2 and must in $V(t)$ near v, again contradicting the fact that the path $g(v,t)$ exits Δ through f and its portion contained in f is a straight line. □

The vertices of $\overline{V^*}$ on ∂U_2 that are neither anchor points nor lie on e are roots as shown by the following Lemma.

Lemma 3.3.7 *Let Δ be a triangle entered through f and let its other two sides be f' and f''. No vertex v of $\overline{\mathcal{V}^*} \cap U_2$ in the interior of f' or f'' has an outgoing edge σ that emanates from v into the interior of Δ.*

Proof: The edge σ in question must intersect (say) f' transversally at v (otherwise it is a degenerate extension segment, and so is an *incoming* edge at v) and, by Lemma 3.3.3 and smoothness of bisectors, the tangent to σ at v must intersect f, making σ an *incoming* edge at v by definition of orientation on σ. □

Note: There may exist internal vertices of $\overline{\mathcal{V}^*}$ that lie on a chord f' separating two triangles Δ and Δ' of U_2 with Δ entered through f and Δ' through f'. In this case, Lemma 3.3.7 shows that there can be no edge leaving v into Δ and reasoning similar to Lemma 3.3.6 shows that there is *exactly* one edge emanating from v into Δ', thereby showing that v has indeed out-degree one and thus is a valid internal vertex of the root-directed forest.

Corollary 3.3.8 *$\overline{\mathcal{V}^*}$ is a root-directed forest with leaves on e and at anchor points, roots at non-anchor vertices of $\mathcal{V}^*$ on $\partial U_2 - e$, all internal non-root nodes being Voronoi vertices and breakpoints and having in-degree at least two.*

Let Δ be a triangle entered through f and let f' and f'' be its two other sides. Define the *sweep-curve* s_τ to be the broken line consisting of a segment parallel to f at distance τ from it extending from f' to f'' together with two portions of f' and f'' from the endpoints of f to the endpoints of that segment. Let r be the distance from f to the vertex of Δ incident to both f' and f''.

Corollary 3.3.9 *With respect to the above convention for orienting the edges of $\overline{\mathcal{V}^*}$, s_τ (for $0 < \tau < r$) separates the vertices of $\overline{\mathcal{V}^*}$ into two sets so that all directed edges connecting two vertices of different sets emanate from below s_τ (i.e., from vertices lying on the side of s_τ that contains f) and terminate above s_τ. The remaining edges do not meet s_τ. Moreover, if τ is such that s_τ passes through a vertex v of $\overline{\mathcal{V}^*}$ in the interior of Δ or f, there is* precisely one *edge leaving v and this edge extends from v into the area* above *s_τ (Figures 3.8a and b).*

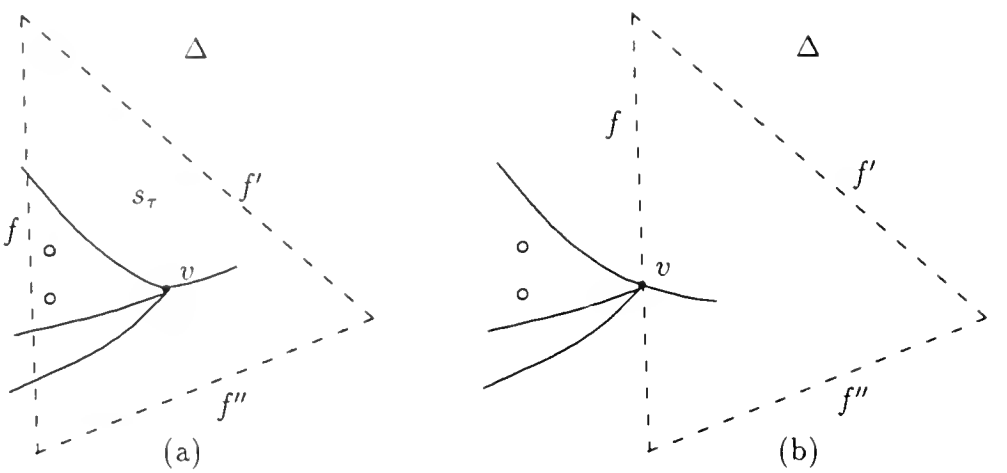


Figure 3.8: Sweeping a triangle.

Lemma 3.3.10 *Given* $\mathcal{V}^* \cap f$, $\mathcal{V}^* \cap \Delta (= \overline{\mathcal{V}^*} \cap \Delta)$ *can be constructed by sweeping* Δ *with* s_τ.

Proof: It is sufficient to systematically locate two (or more) children with a common parent in $\overline{\mathcal{V}^*} \cap \Delta$, and replace the children by the parent, thereby advancing the sweep-curve through the forest one step. This in turn reduces to maintaining the intersection of $\mathcal{V}^*$ with the current sweep-curve and repeatedly locating the pair of adjacent edges whose intersection point lies on s_τ with least τ. □

We are now ready to describe the algorithm for extending $\mathcal{V}^*_{U_1}(S)$ to U_2. Throughout the algorithm $\mathcal{V}^*_{U_{cur}}$ for the current polygon U_{cur} is maintained while U_{cur} grows from U_1 to U. More precisely, maintain the planar embedding of $\partial U_{cur} \cup \mathcal{V}^*_{U_{cur}}(S)$. Assume that the planar map for U_1 is available initially. Since possessing $\mathcal{V}^*_{U_1}(S)$ implies knowledge of the sorted order in which edges of $\mathcal{V}^*$ meet the chord e separating U_2 from U_1, we can build in linear time a search tree containing this information. More precisely, the tree will represent those refined Voronoi cells of $\mathcal{V}^*_{U_1}(S)$ that are adjacent to e. Namely, it will contain for each such cell $V^*_a(s)$ its *owner* s, its *anchor* a, and the *weight* $d(s,a)$ of the anchor. The tree must support *insert*, *delete* and

split operations in time logarithmic in its maximum size (one can use, for example, red-black trees of [GS78]).

The algorithm for constructing $\mathcal{V}_U^*(S)$ from $\mathcal{V}_{U_1}^*(S)$ will proceed by performing some initial setup at the edge e and then traversing the triangulation tree of U_2 starting from the triangle adjacent to e:

Algorithm 3.2

Input: A triangulated polygon U cut by a chord e of the triangulation into subpolygons U_1 and U_2, and the planar map representation of $\mathcal{V}_{U_1}^*(S) \cup \partial U_1$.

Output: The planar map representation of $\mathcal{V}_U^*(S) \cup \partial U$.

1. Initial setup

   (i) From $\mathcal{V}_{U_1}^*(S)$, extract the ordered list L of refined Voronoi cells adjacent to e.

   (ii) Convert the list into a search tree T.

   (iii) For each pair of adjacent refined Voronoi edges bounding a single refined Voronoi cell on L (and thus in T) compute their point of intersection (a *candidate*) without checking whether the resulting candidates are feasible in any sense. Perform a point location query on each of the candidates and collect them into buckets corresponding to the triangle where each of them lands (an equally feasible approach would be to prioritize the buckets immediately).

2. Process every triangle of the triangulation of U_2 in a pre-order traversal of its triangulation tree, starting at the triangle adjacent to e and treating the tree as rooted at that triangle. The order in which children of a triangle are visited is immaterial. Upon entering, through edge f, triangle Δ, whose associated tree T represents the refined Voronoi cells of $\mathcal{V}_{U_{cur}}^*(S)$ adjacent to f (or, equivalently, the refined Voronoi cells of $\mathcal{V}_U^*(S)$ intersecting f), perform:

(i) Let x be an endpoint of f and y be the anchor of the refined Voronoi cell that contains x. Consider the ray out of x directed away from y. If this ray partially overlaps f, create a degenerate refined partition edge that emanates from x and follows f until its first intersection with another edge of $\mathcal{V}^*$ as represented by T. If the said ray intersects the interior of Δ, create a refined partition edge emanating from x and directed into Δ. In either case, add the intersection of the new refined partition edge with its neighbor in T to the bucket of the appropriate triangle and create a new region sandwiched between the new refined partition edge and the second edge of Δ incident to x with its owner being the owner of y, its anchor being x and the weight of its anchor being the sum of $d(x, y)$ and the weight of y. Otherwise there is no need to create a new refined partition edge or a new region. Repeat the same procedure for the other endpoint of f. Note that creation of a region requires an insertion into T.

(ii) Construct a priority queue Q containing all the candidates in the bucket associated with Δ. Q is ordered by increasing distance from the line containing f, which amounts to sweeping across Δ with s_τ.

(iii) The candidate intersection p in Q closest to f is then repeatedly deleted from Q. If p is actually valid, i.e., it refers to an intersection of two curves currently adjacent on s_τ (i.e., bounding a common refined Voronoi cell in T), it must correspond to the disappearance of the cell between them from T and the replacement of the two refined bisector edges (or a refined bisector edge and a refined partition edge) by a new refined bisector edge. At this point the region is deleted from T. If p does not represent a valid intersection, it is simply discarded. The two pairs of curves newly adjacent on s_τ have their intersections computed and located in the triangulation. If they land in Δ, they are added to Q; otherwise, they are added to the appropriate buckets. The process repeats until Q is empty.

(iv) Let f' and f'' be the remaining edges of Δ and z be their common endpoint.

When the construction inside Δ is completed, the two trees corresponding to f' and f'' are produced by splitting the current version of T at z, conceptually cutting s_τ at that point. Note that s_τ coincides with $f' \cup f''$ at this point, so splitting T correctly creates the two initial trees representing s_0 for the two unvisited triangles adjacent to Δ. In case f' (resp. f'') is actually an edge of U, the corresponding search tree is converted to adjacency information for $\mathcal{V}^*$ along it.

We proceed to analyze the time complexity of Algorithm 3.2. Steps 1(i) and 1(ii) clearly take linear time. As there is only a linear number of initial candidate pairs of adjacent curves on s_τ, step 1(iii) requires $O((n+k)\log n)$ time. One can easily see that the time complexity of part 2 is $O(\log n)$ per candidate intersection for point location, $O(\log(n+k))$ for queue maintenance operations (as no candidate is ever in two queues), $O(\log(n+k))$ for tree deletion for each *valid* intersection (i.e., a Voronoi vertex or a breakpoint), and $O(\log(n+k))$ for tree insertion and split per vertex of U_2. The last three bounds hold since the size of all queues and trees is at all times $O(n+k)$, which we argue as follows. In fact, each candidate intersection is either created initially (there are only $O(|L|) = O(n+k)$ of such) or added during a deletion from or insertion into the tree (at most two new candidates per insertion, one new candidate per deletion). There are at most n insertions and $O(n+k)$ deletions–note that the universe does not necessarily decrease all the time; in fact it can grow by as much as n due to introduction of new refined partition edges, which can occur at most once per reflex corner of U_2. Thus the *total* size of all queues and trees at every point of the algorithm is bounded by $O(n+k)$, as claimed. In other words, part 2 spends logarithmic time per candidate intersection which, there being $O(n+k)$ candidate intersections, bounds the execution time of part 2 by $O((n+k)\log(n+k))$.

Therefore the extension of the refined Voronoi diagram of a set of k points from a portion of an n-gon to the whole polygon is accomplished in $O((n+k)\log(n+k))$ time by Algorithm 3.2, providing the desired bound for part 2(v) of Algorithm 3.1. Observe that it is impossible to improve Algorithm 3.2 so that it runs in time $O(n+k)$ without modifying the remainder of Algorithm 3.1, for such an improvement would reduce the time complexity of the entire algorithm to $O((n+k)\log n)$, contradicting the lower bound discussed in Section 4.3.

3.3.2 Merging Two Diagrams

In this Section we describe a linear-time algorithm for merging the (closest-site geodesic) Voronoi diagrams of two sets of sites in a polygon assuming that a chord

of the polygon separates the two sets. Again, before detailing the algorithm, we will examine some properties of Voronoi diagrams of such sets of sites.

Let A be a set of sites in U and x be a point in U. Then the *distance from A to x* is defined by $d_A(x) = \min_{s \in A} d(s, x)$. For $A, B \subset U$ disjoint non-empty sets of sites, define the *bisector of A and B* to be $b(A, B) = \{x \in U \mid d_A(x) = d_B(x)\}$ and let $H(A, B) = \{x \in U \mid d_A(x) < d_B(x)\}$. Trivially $\{H(A, B), b(A, B), H(B, A)\}$ is a partition of U. By continuity of $\lambda(x) = d_A(x) - d_B(x)$, $b(A, B)$ is the relative boundary of both $H(A, B)$ and $H(B, A)$ and the two sets cannot be connected by a path that does not meet $b(A, B)$.

It is easily verified that $b(A, B)$ is the union of all Voronoi edges $e(s, t)$ of $\mathcal{V}_U(A \cup B)$ with $s \in A$ and $t \in B$. In particular, $b(A, B)$ has complexity linear in the sum of sizes of A, B, and U since it is (the union of) a collection of Voronoi edges. Hence $b(A, B)$ can be viewed a subgraph of $\mathcal{V}(A \cup B)$. Moreover, $b(A, B)$ cannot terminate at Voronoi vertices in the interior of U or contain isolated Voronoi vertices since. if a Voronoi vertex v belongs to $b(A, B)$, v is on the relative boundary of Voronoi cells of at least one A-site and at least one B-site. This in turn implies the existence of at least two (and at least one for $v \in \partial U$) Voronoi edges emanating from v that separate cells of sites from different sets. (In fact, this argument shows that an arbitrary Voronoi vertex $v \in b(A, B)$ in the interior of U has even degree in $b(A, B)$. Thus $b(A, B)$ can be decomposed into edge-disjoint simple paths and cycles.)

Lemma 3.3.11 *Suppose $s \in A$. Let $V(s)$ denote the Voronoi cell of s in $\mathcal{V}_U(A)$ and $V'(s)$ denote its cell in $\mathcal{V}_U(A \cup B)$. Let $V_a^*(s)$ and $V_a'^*(s)$ be the corresponding refined Voronoi cells with anchor a. Then*

$$V'(s) = V(s) \cap H(A, B) \quad \textit{and} \quad V_a'^*(s) = V_a^*(s) \cap H(A, B).$$

Proof:

$$\begin{aligned} V(s) \cap H(A, B) &= V(s) \cap \{x \mid d_A(x) < d_B(x)\} \\ &= \{x \mid d(s, x) < \min_{r \in A - \{s\}} d(r, x) \;\&\; \min_{r \in A} d(r, x) < \min_{r \in B} d(r, x)\} \end{aligned}$$

$$= \{x \mid d(s,x) < \min_{r \in A \cup B - \{s\}} d(r,x)\} = V'(s).$$

For refined cells, the claim follows immediately as $V_a^*(s) = V(s) \cap P_a(s)$ and $V_a'^*(s) = V'(s) \cap P_a(s)$. □

Observe that, since $V_a'^*(s)$ is path-connected, it is simply the path-connected component of $V_a^*(s) - b(A,B)$ containing s.

Lemma 3.3.12 *Given $\mathcal{V}_U^*(A)$ and $\mathcal{V}_U^*(B)$ and a point on every connected component of $b(A,B)$, $\mathcal{V}_U^*(A \cup B)$ can be constructed in time $O(|A| + |B| + n)$ where n is the number of corners of U.*

Proof: From each point given on $b(A,B)$ build connected components of $b(A,B)$ by the usual Shamos-Hoey scan of the two Voronoi diagrams (see, for example, [Lee78] or [SH75]). One can easily follow a bisector of two sites as long as the two anchor points stay fixed. The anchor points change precisely when the bisector crosses a refined partition edge, thereby moving from one refined Voronoi cells to another (belonging to the same site). At this point one can recover the new anchor point as it is simply the anchor associated with the refined Voronoi cell being entered. The fine structure of the Voronoi cells (namely the refined Voronoi diagram) is somewhat reminiscent of the "spokes" of Kirkpatrick [Kir79] (see below). A complication arises when vertices of degree greater than two are encountered on $b(A,B)$. At such a Voronoi vertex v the list of owners and anchors of adjacent refined Voronoi cells in both diagrams is known, allowing one to determine *all* the edges leaving v that separate a cell of an A-site from that of a B-site, thus enabling the Shamos-Hoey scan to trace *all* branches of $b(A,B)$. The additional effort required is proportional to degree of v in $\mathcal{V}_U(A \cup B)$, thus it is linear over the construction of $b(A,B)$. Note that this complication could be avoided completely by assuming (as it is common in many Voronoi diagram algorithms) that no point is simultaneously equidistant from four sites, thus excluding Voronoi vertices of degree larger than three and ensuring degree at most two for every vertex of $b(A,B)$.

Now split both refined Voronoi diagrams along $b(A,B)$ and collect the portions of refined Voronoi cells reachable from their owners; the resulting collection is (by

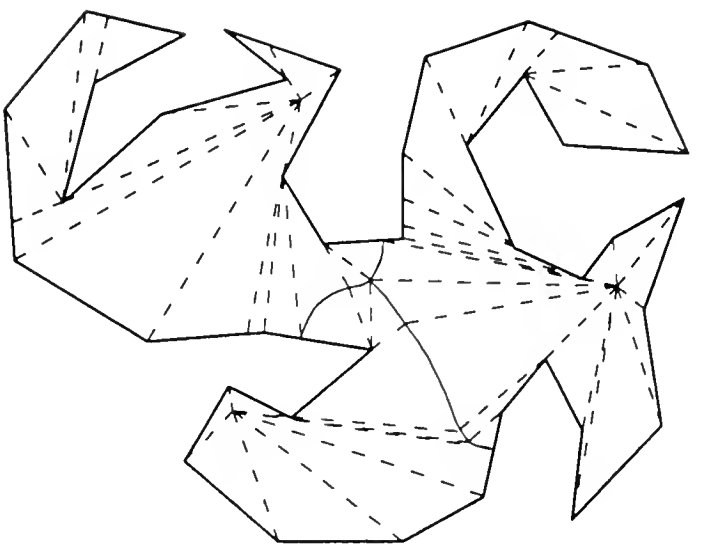


Figure 3.9: Voronoi triangulation for the three sites of Figure 3.1.

Lemma 3.3.11) the refined Voronoi diagram of $A \cup B$.

To repeatedly locate the intersection of a refined bisector edge with the boundary of "current" refined Voronoi cells of $\mathcal{V}_U^*(A)$ and $\mathcal{V}_U^*(B)$ in an efficient manner, we follow Kirkpatrick [Kir79] and introduce a further refinement of our map that will guarantee that each individual region has bounded complexity, thereby allowing us to trace a refined bisector edge within each such region in constant time. In this extra refinement, in addition to the refined partition edges, every vertex of ∂U and every Voronoi vertex lying on the boundary of a Voronoi cell are connected by a shortest path to the owner of the cell (actually, one need only connect each such point to its anchor, so the size of the resulting structure is still linear). The structure described coincides with the "Voronoi triangulation" of [AA87] (Figure 3.9). Moreover, since refined Voronoi cells are star-shaped and each added segment is contained in the shortest path to the owner of the cell, it will not intersect $b(A, B)$ more than once. It is not difficult to see that every face of the resulting map is bounded by two segments and (in general) a hyperbolic arc or a polygon boundary segment, and thus

has complexity bounded by a constant. It is easy to verify that this finer structure allows Kirkpatrick's tracing procedure to run in linear time. □

Note: Observe that the two steps described above need not be done separately. In fact, it is more convenient to perform truncation of refined Voronoi cells by $b(A, B)$ at the time it is being constructed, concurrently updating the two Voronoi diagrams and merging them along $b(A, B)$.

Corollary 3.3.13 *If all components of $b(A, B)$ meet ∂U, $\mathcal{V}_U^*(A \cup B)$ can be produced from $\mathcal{V}_U^*(A)$ and $\mathcal{V}_U^*(B)$ in time $O(|A| + |B| + n)$.*

Proof: It is sufficient to compute $b(A, B) \cap \partial U$. Since both $\mathcal{V}_U^*(A)$ and $\mathcal{V}_U^*(B)$ are linear in size, it is possible to trace ∂U in linear time, partitioning it into $O(n + k)$ line segments each of which lies in exactly one refined Voronoi cell of each $\mathcal{V}_U^*(A)$ and $\mathcal{V}_U^*(B)$. Each segment contains points not only nearest to a unique $s \in A$ and a unique $t \in B$ but also having a unique pair of anchors $(\hat{s}, \hat{t})$ with $d(s, \hat{s})$ and $d(t, \hat{t})$ known. Thus, on each segment, the analytic expressions for both

$$d_A(x) = d(s, x) = d(\hat{s}, s) + |x - \hat{s}|$$

and

$$d_B(x) = d(t, x) = d(\hat{t}, t) + |x - \hat{t}|$$

are known. It is easily verified that the two functions coincide in at most two points, which can be computed in constant time. Each such point corresponds to an intersection of $b(A, B)$ and ∂U, thereby enabling us to compute *all* such intersections in linear time and permitting the application of Lemma 3.3.12.

Note that the case when the functions $d_A(x)$ and $d_B(x)$ coincide violates the general position assumption as in such a case $\hat{s} = \hat{t}$. □

Lemma 3.3.14 *If A and B are disjoint sets of sites in U separated by a boundary geodesic $g(u, v)$, $b(A, B)$ contains no cycles. In particular, the conditions of Corollary 3.3.13 are satisfied.*

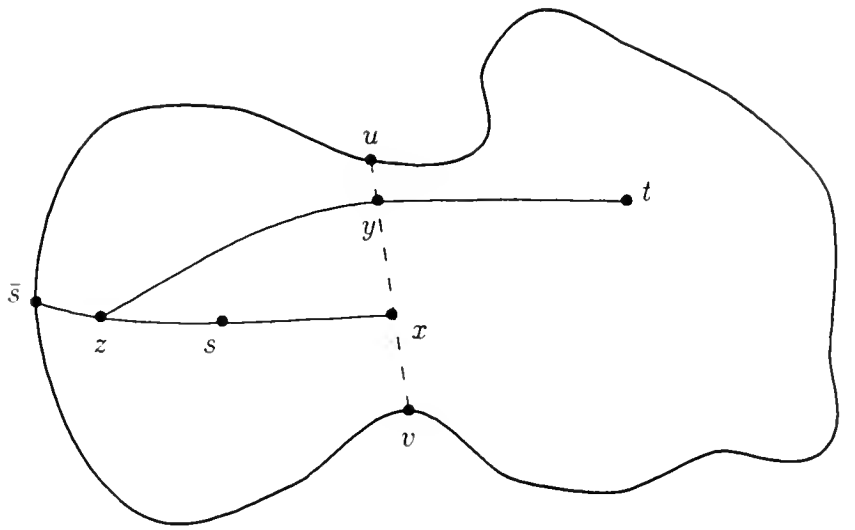


Figure 3.10: Illustration to the proof of Lemma 3.3.14.

Proof: Let $u, v \in \partial U$ such that $A \subseteq U[u,v]$ and $B \subseteq U[v,u]$. Notice that $H(A,B)$ is the union of Voronoi cells of A-sites (in $\mathcal{V}_U(A \cup B)$) together with the Voronoi edges separating these cells (an endpoint of such an edge lies in $H(A,B)$ if it is incident only to the cells of A-sites, otherwise it lies on $b(A,B)$). Hence, if there is a cycle γ in $b(A,B)$, it must completely enclose a cell of, say, A-site s, so that there is no path in $H(A,B)$ connecting s to ∂U. First assume $s \notin g(u,v)$. Choose x to be the point of $g(u,v)$ closest to s. Necessarily either $x \neq u, v$ and $m\angle uxs, m\angle sxv \geq \pi/2$ or $x = u$ (resp. v) and $m\angle sxv \geq \pi/2$ (resp. $m\angle uxs \geq \pi/2$). Let $\bar{s} \in \partial\mathcal{V}$ be a shadow of $g(x,s)$ (the situation is schematically depicted in Figure 3.10). We claim that $g(s,\bar{s}) \subset H(A,B)$, contrary to our assumption that s is enclosed by a cycle $\gamma \subset b(A,B)$. Consider $z \in g(s,\bar{s})$ and $t \in B \subset U[v,u]$. If $x \in g(z,t)$, $d_z(t) \geq d_z(x) \geq d_z(s)$ (note that $s \neq t$, since A and B are disjoint), so $d_z(t) \geq d_z(s)$. Let us assume $x \notin g(z,t)$. Let $y \in g(u,v) \cap g(z,t)$. Notice that $x \neq y$ and, by the above observation, the angle between $\theta(x,z)$ $(= \theta(x,s))$ and $\theta(x,y)$ has measure $\geq \pi/2$. By Lemma 2.2.2, $d(z,y) > \max\{d(x,y), d(x,z)\}$. Thus $d_z(t) \geq d_z(y) > d_z(x) \geq d_z(s)$.

Hence $d(z, B) > d(z, s)$, and therefore $z \in H(A, B)$, as asserted.

The case $s \in g(u, v)$ is handled similarly. As $s \notin \partial U$ by assumption, we can choose $\bar{s}$ to be the first intersection of $\partial U[u, v]$ with the ray emanating from s into $U[u, v]$ perpendicular to $\theta(s, u)$. The remainder of the argument is identical to the previous case. □

Corollary 3.3.15 *The diagrams $\mathcal{V}_U^*(S_L)$ and $\mathcal{V}_U^*(S_R)$ in step 2(vi) of Algorithm 3.1 can be merged in time $O(n + k)$.*

To summarize, the procedure for merging the two diagrams consists of locating all points of $b(A, B)$ on ∂U by a linear scan followed by tracing of every connected component of $b(A, B)$ *a la* Kirkpatrick [Kir79].

3.4 A Special Case

In this Section we will consider a variation of our algorithm that computes the geodesic Voronoi diagram for k point sites among which are *all* reflex corners of the enclosing n-gon, in $O((n + k)\log(n + k))$ time. This variant exploits the following property of the geodesic Voronoi diagram of such a set of sites–the Voronoi cells are star-shaped in the Euclidean metric, as a bend on a path from a point in a cell to the owner of the cell would produce a reflex vertex, i.e., a closer site. The general strategy is still divide-and-conquer, with the polygon recursively split into two roughly equal parts and the resulting diagrams merged *a la* Shamos and Hoey [SH75] (or Kirkpatrick [Kir79]). However, the extension step is avoided, rather a structure similar to the extension of Voronoi diagram into "site-less" regions is built at the bottom of recursion and maintained together with the Voronoi diagram proper during the merging steps. The rationale behind this approach is that the only sites that can own a given point of the polygon are those visible from it. Consequently, during the merge step, the only points of one half of the polygon that can be "taken over" by sites of the other half are those visible from the sites through the "window" of the dividing chord. In particular,

it is enough to construct the Voronoi diagram of the set of sites of one half of the polygon as visible through this window. This notion is identical to the notion of the "peeper's Voronoi diagram" of [BS88], but fortunately in our case it is easily seen to have linear size. Moreover, we can show that the geodesic Voronoi diagram of a set of sites in a polygon with "peeper's Voronoi diagrams" attached to some of the polygon edges (in fact, to those which are chords of triangulation of the original polygon) is a structure linear in the size of the polygon and the number of sites and that two such structures can be merged in linear time. This yields an $O((n+k)\log(n+k))$ algorithm for constructing the geodesic Voronoi diagram in this special case. We omit the details of this faster algorithm.

Note that the algorithm sketched above allows, in particular, to compute the geodesic Voronoi diagram of the *vertices* of a simple n-gon in time $O(n \log n)$. An alternative $O(n \log n)$ algorithm is provided by work of Lee and Lin [LL86] (see also [Che89] and [WS87]), which involves the calculation of the *generalized Delaunay triangulation* of a simple polygon U. It is defined as a triangulation of the polygon with the property that the circle circumscribed around each face Δ of it does not contain (in its interior) any vertex of U *visible* simultaneously from all three vertices of Δ. In the case of a convex polygon, this definition gives precisely the conventional Delaunay triangulation of U, which is the dual of the (conventional) Voronoi diagram of the vertices of U, which in turn coincides (inside of U) with the geodesic Voronoi diagram of the vertices of U. In fact, it is not difficult to observe that, for a general simple polygon U, the generalized Delaunay triangulation is essentially the dual of the geodesic Voronoi diagram (in fact, it can be shown to have more edges than the dual, intuitively because there are Voronoi edges that do not appear in the diagram because of the boundedness of U). Lee and Lin [LL86] (and also Chew [Che89], Wang and Schubert [WS87]) provide an $O(n \log n)$ algorithm for constructing this triangulation, and the diagram can be reconstructed from it in linear time by visiting the neighbors of each vertex in cyclical order and thereby reconstructing its Voronoi cell. Thus we obtain an alternative $O(n \log n)$ algorithm for computing the geodesic Voronoi

diagram of the set of vertices of a simple polygon (with respect to that polygon).

Chapter 4

Furthest-Site Geodesic Voronoi Diagram

In this Chapter, we come to another classic structure of Euclidean geometry, namely the "furthest-site Voronoi diagram." Given a finite collection of sites in the plane, the furthest-site Voronoi diagram partitions the plane into Voronoi cells, one cell per site. The site that owns a cell is the site that is furthest from every point in the cell. Using well-known algorithms, the Euclidean furthest-site Voronoi diagram of k point sites can be computed in time $O(k \log k)$ and space $O(k)$ [PS85].

The content of this Chapter is an efficient algorithm for computing the furthest-site Voronoi diagram, defined by the geodesic metric inside a simple polygon. The algorithm uses $O((n+k)\log(n+k))$ time and $O(n+k)$ space, where n is the number of bounding edges of the polygon and k is the number of (point) sites. The best previous algorithm for this problem has running time $O(n^3 \log\log n)$ [AT86], and just computed (a superset of) the vertices of the furthest-site Voronoi diagram of the n corners of the polygon. We remark that our furthest-site geodesic Voronoi diagram algorithm is a factor of $O(\log n)$ faster than the best known *nearest*-site geodesic Voronoi diagram algorithm (cf. Chapter 3).

The problem of computing the furthest-site Voronoi diagram is an extension of the "furthest neighbor problem," which is "Given a finite collection of points, for

each point identify the element in the collection that is maximally distant from it." Suri [Sur87] shows how to solve a special case of the furthest neighbor problem in the geodesic metric inside a simple polygon. Specifically, he gives an algorithm that for each corner of the polygon computes the corner that is maximally distant from it in the geodesic metric. His algorithm runs in time $O(n \log n)$ and space $O(n)$, where n is the number of bounding edges of the polygon.

The furthest-site geodesic Voronoi diagram generalizes the notion of the geodesic furthest-neighbor mapping of Suri [Sur87] in two ways. First, the Voronoi diagram provides a planar partition of the polygon together with its interior into furthest-site Voronoi cells. Consequently, arbitrary furthest-site queries can be answered using a planar point-location algorithm. Second, the set of sites is not restricted to the corners of the polygon. Rather, the sites can be arbitrarily situated in the polygon. Both of these generalizations have substantial technical impact on the algorithm for computing furthest-site Voronoi diagrams.

There are many analogies between the Euclidean furthest-site Voronoi diagram and the geodesic furthest-site Voronoi diagram. In the Euclidean case, if a site has nonempty Voronoi cell, then it lies on the convex hull of the set of sites and it does not appear on the line segment between two other sites. The counterclockwise sequence of Voronoi cells (at infinity) is the same as the counterclockwise sequence of sites on the convex hull. In the geodesic case, we show that a site with nonempty Voronoi cell lies on the relative convex hull of the set of sites and does not appear on a geodesic between two other sites. The counterclockwise order of Voronoi cells along the boundary of the polygon is consistent with the counterclockwise order of sites on the relative convex hull. In the Euclidean case this characterization is exact: any site on the convex hull not between two other sites has nonempty Voronoi cell. In the geodesic case the characterization is not exact, roughly because the polygon may not be large enough for the cell to appear.

A further analogy between the two cases is the structure of the Voronoi diagram itself. In the Euclidean case the Voronoi diagram forms a tree with root at the

Euclidean center of the set of sites. (The center of a set of point sites is the point that minimizes the maximum distance to any site.) If edges are directed towards the root, then this orientation is consistent with geometric direction towards the center. In the geodesic case exactly the same properties hold, substituting "geodesic center" for "Euclidean center" and "geodesic direction" for "direction."

The algorithm for computing the furthest-site Voronoi diagram consists of two steps. First, we compute the restriction of the Voronoi diagram to the boundary of the polygon. Intuitively, the boundary of the polygon in the geodesic case corresponds to points "at infinity" in the Euclidean case. Second, we extend the diagram to the interior of the polygon. Since the Voronoi diagram forms a tree with root at the geodesic center, the second step is relatively easy. It can be performed by a "reverse geodesic sweep" towards the geodesic center.

The first step, the computation of the Voronoi diagram on the boundary of the polygon, is much more involved. We use a technique developed by Suri [Sur87] for determining furthest neighbors. We reduce the problem to three instances of the "two-fragment problem"; an instance of the two-fragment problem consists of a fragment of the boundary of the polygon and a fragment of the relative convex hull of the set of sites. The relative convex hull fragment contains the furthest sites of all points on the polygon boundary fragment. We solve an instance of the two-fragment problem using divide and conquer in the following manner. The polygon boundary fragment is split at its "midpoint," leaving roughly equal number of corners on either side; this implies a corresponding split of the convex hull fragment. Thus one instance of the two-fragment problem results in two simpler instances. Eventually instances become small enough to be solved directly.

We refine Suri's two-fragment technique in two ways. First, Suri's algorithm always splits the polygon boundary fragment at a corner of the polygon. This is sufficient for the furthest-neighbor problem, because furthest-neighbor information for points along a wall is not of interest. For the furthest-site Voronoi diagram, simply splitting at corners is insufficient, since potentially many Voronoi cells may meet a

single wall of the polygon. If necessary, we further split each wall into subsegments so that the shortest path tree from a point in a subsegment to the sites is combinatorially invariant over the entire subsegment. The combinatorial invariance of the shortest path tree implies that the Voronoi partition of the subsegment can be easily computed.

The second refinement of Suri's technique concerns the complexity analysis of the recursion. Suri's original algorithm required a step called "trimming"; trimming a two-fragment instance introduces a different subproblem that could be solved directly. This operation is necessary in Suri's analysis in order to maintain the linearity of the total size of all subproblems at a particular level of recursion. We show that even without trimming, the total size of all subproblems at a particular level is linear. This observation simplifies the recursive structure of the "two-fragment" algorithm so that it actually matches the description given above. The analysis has also been incorporated into the final version of Suri's furthest-neighbor algorithm.

The best lower bound that we know for computing the furthest-site geodesic Voronoi diagram is $\Omega(n + k \log k)$ (cf. Section 4.3). It follows quite easily from known lower bounds for diameter computation in the Euclidean case. Conceivably, the current algorithm could be improved to match this lower bound.

4.1 More Geometry

This Section contains the definition of the furthest-point geodesic Voronoi diagram and some of its basic properties. Some of the notation of Chapter 3 is intentionally reused, where possible, to emphasize similarities in the properties of the closest-site and the furthest-site geodesic Voronoi diagrams. We begin by defining the geodesic furthest-site Voronoi diagram in Section 4.1.1. Section 4.1.2 contains the Ordering Lemma, which states that the order of Voronoi cells around the boundary of the containing polygon is consistent with the order of sites around the relative convex hull of the set of sites. In Section 4.1.3 we define a refined form of the Voronoi diagram and use it to show a linear bound on the descriptive complexity of the (unrefined)

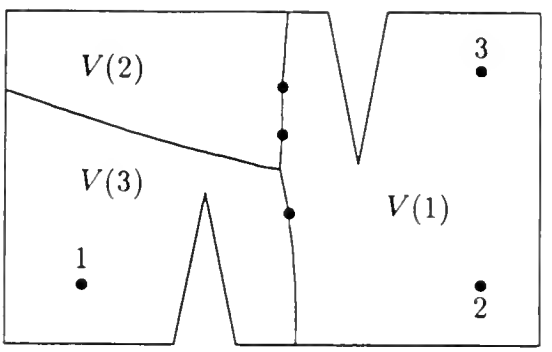


Figure 4.1: Voronoi cells of the sites of Figure 2.10.

diagram. Finally, in Section 4.1.4 we show that the Voronoi diagram forms a tree directed towards the geodesic center of the set of sites. The algorithm for computing the diagram is given in Section 4.2.

4.1.1 Voronoi Cells

Recall that S is the set of sites in the universe U. The *(geodesic furthest-site) Voronoi cell* of site s is $V(s) = \bigcap_{t \neq s} H(t,s)$. The *(geodesic furthest-site) Voronoi diagram* $\mathcal{V}$ is

$$\{x \in b(s,t) \mid s,t \in S \text{ and } d_s(x) = \max_{r \in S} d_r(x)\}.$$

Figure 4.1 indicates the Voronoi diagram and Voronoi cells of the three points depicted in Figure 2.10. A *Voronoi edge* $e(s,t)$ is

$$e(s,t) = \mathcal{V} \cap b(s,t) = b(s,t) \cap \bigcap_{r \neq s,t} (H(r,s) \cup b(r,s)).$$

If the intersection consists of more than one point (else we will say that $e(s,t)$ does not exist). A *(Voronoi) vertex* is a point $x \in \mathcal{V}$ which has three or more sites furthest from it. By Lemma 2.5.4, there is at most one such point x for each triple of sites. A *hitpoint* is the intersection of a Voronoi edge with ∂U. Intuitively, a hitpoint corresponds to the "point at infinity" of an infinite Voronoi edge in a Euclidean furthest-site Voronoi diagram.

The proof of the following two Lemmas is essentially identical to that of Lemmas 3.1.1 and 3.1.2:

Lemma 4.1.1 *Each Voronoi edge $e(s,t)$ is connected and has vertices or hitpoints as endpoints.*

Lemma 4.1.2 *Suppose $s_1, ..., s_h, s_{h+1} = s_1$ are the sites furthest (and thus equidistant) from vertex v, and directions $\theta(v,s_1), ..., \theta(v,s_h)$ are in counterclockwise order. Then, for each i, edge $e(s_i, s_{i+1})$ is incident to v and extends away from v in direction bisecting $\angle s_{i+1} v s_i$, as long as that direction (locally) stays inside U.*

If vertex v appears on ∂U, then there is only one edge of $\mathcal{V}$ incident to v: as v cannot be a corner of ∂U by the general position assumption, it must be an interior point of a wall. Hence only the edge bisecting $\angle svt$ remains within U, where directions $\theta(v,s)$ and $\theta(v,t)$ are the most clockwise and most counterclockwise directions towards sites furthest from v, respectively.

The following Lemma is in some sense dual to the fact that both the closest-site geodesic Voronoi cell and its closure are star-shaped with respect to its owner site (Lemma 3.1.3 of Section 3.1).

Lemma 4.1.3 (Extension Lemma) *If x lies on $g(s,y)$ and $x \in \mathrm{Cl}(V(s))$, then all of $g(s,y)$ past x lies in $V(s)$.*

Proof: Immediate from the definition of a Voronoi cell and Lemma 2.6.1(2). □

It is an immediate consequence of this Lemma that every point in a Voronoi cell is connected to ∂U: if $x \in V(s)$, then segment $x\bar{x} \subseteq V(s)$, where $\bar{x}$ is a shadow of $g(s,x)$.

Lemma 4.1.4 *Both $V(s) \cap \partial U$ and $V(s)$ are path-connected.*

Proof: Since every point of $V(s)$ is connected to a point of $V(s) \cap \partial U$, it suffices to show $V(s) \cap \partial U$ is connected. To prove it, we will show that $H(r,s) \cap H(t,s) \cap \partial U$

is connected for every $r, t \neq s$. Label the hitpoints of $b(r,s)$ and $b(t,s)$ as x_r, y_r and x_t, y_t, respectively, so that $(\partial U \cap H(r,s)) \cup \{x_r, y_r\} = \partial U[x_r, y_r]$ and $(\partial U \cap H(t,s)) \cup \{x_t, y_t\} = \partial U[x_t, y_t]$. The only counterclockwise ordering of these points that disconnects $\partial U \cap H(r,s) \cap H(t,s)$ is x_r, y_t, x_t, y_r. In particular, this implies that $(H(r,s) \cup b(r,s)) \cup (H(t,s) \cup b(t,s))$ cover ∂U. We show this is impossible; suppose this were the ordering. Since $s \notin H(r,s) \cup H(t,s)$, $H(r,s) \cup H(t,s) \neq U$, so $b(r,s)$ must intersect $b(t,s)$ in at least two distinct points. But this contradicts Lemma 2.5.4, since each of the two points would be equidistant from sites r, s, and t. □

4.1.2 The Ordering Lemma

Let C be the relative convex hull of S, the set of sites. The first Lemma of this Section shows that, for $V(s)$ not to be empty, s must be an extreme point of S (and in fact on the far side of S from any point in the closure of $V(s) \cap \partial U$). Hence we can assume that all sites are extreme and that they are ordered by the counterclockwise traversal of ∂C. The main result of this Section is the Ordering Lemma: the order of Voronoi cells around ∂U is consistent with the order of sites around ∂C. In addition we show that there is a collection of at most three geodesics separating every point of ∂U from its furthest site (or sites). This is used to prove the rather remarkable fact that there are at most $O(n+k)$ distinct links in all geodesics that connect corners of ∂U to their respective furthest sites. We remark that Lemmas 4.1.6, 4.1.9, and 4.1.10 are based on very similar Lemmas proved by Suri [Sur87].

Lemma 4.1.5 *If $V(s)$ is not empty, then s is an extreme point of S on the far side of C from any x in the closure of $V(s) \cap \partial U$.*

Proof: Suppose s is a site furthest from x. If $s \in g(r,t)$, for r, t sites distinct from s, then by Lemma 2.2.1, $d_s(x) < \max\{d_t(x), d_r(x)\}$, a contradiction. Suppose the last segment of $g(x,s)$ can be extended beyond s staying in C; let s' be a point lying on such an extension and on the boundary of C. Then $d_{s'}(x) > d_s(x)$, s' must lie on $g(t,u)$ for some sites t, u, and $d_{s'}(x) \leq \max\{d_t(x), d_u(x)\}$, contradicting the choice

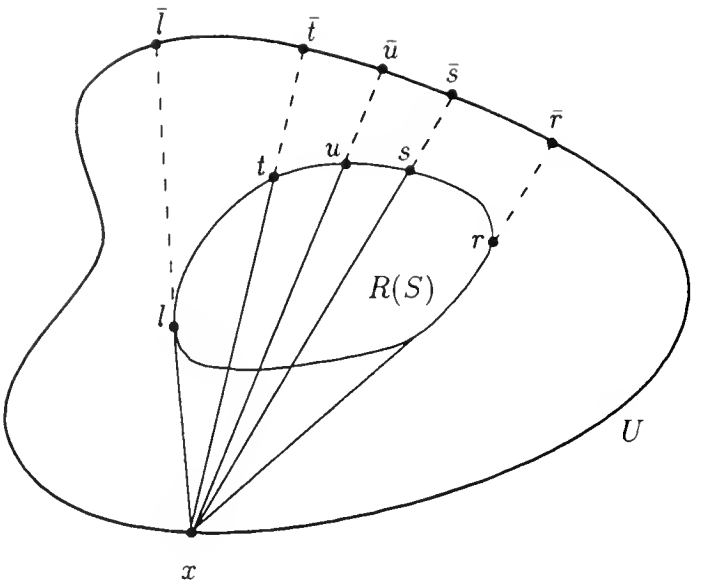


Figure 4.2: Illustration to the proof of Lemma 4.1.7

of s. Hence s is not an interior point of C, so s must be an extreme point. By Lemma 2.4.1, s is on the far side of C from x. □

Lemma 4.1.6 *Suppose s and t are furthest sites from $u, v \in U$, respectively, with $u \neq v$, $s \neq t$. Then $g(u,t)$ does not meet $g(v,s)$.*

Proof: Immediate from Lemma 2.5.2. □

Lemma 4.1.7 *Suppose $V(t) \cap \partial U$ immediately follows $V(s) \cap \partial U$ in counterclockwise order around ∂U. If $u \neq s,t$ is another site on $\partial C[s,t]$, then $V(u)$ is empty.*

Proof: Let x be the hitpoint of $e(s,t)$ so that (near x) $V(s)$, x, $V(t)$ are in counterclockwise order along ∂U. Let $\bar{s}$ and $\bar{t}$ be closest shadows of $g(x,s)$ and $g(x,t)$, respectively. (For a schematic diagram, refer to Figure 4.2.) We show that x, $\bar{s}$, $\bar{t}$ are distinct and appear in that order counterclockwise around ∂U. Note that by

Lemma 2.5.1, $g(x,\bar{s}) = \bar{g}(x,s)$ and $g(x,\bar{t}) = \bar{g}(x,t)$ are geodesics emanating from x with distinct initial directions; hence $\bar{s} \neq \bar{t}$. Near x, $V(t) \cap \partial U = H(s,t) \cap \partial U$ and $V(s) \cap \partial U = H(t,s) \cap \partial U$; also $\theta(x,s)$ enters $H(s,t)$ and $\theta(x,t)$ enters $H(t,s)$. Since $\bar{g}(x,s)$ does not intersect $\bar{g}(x,t)$ again, the ordering of x, $\bar{s}$, $\bar{t}$ must be counterclockwise around ∂U.

We claim $\bar{s}$, $\bar{u}$, and $\bar{t}$ are in that counterclockwise order on ∂U, where $\bar{u}$ is the closest shadow of $g(x,u)$. Let r and l be the clockwise and counterclockwise extreme points of S from x, with $\bar{r}$ and $\bar{l}$ closest shadows of $g(x,r)$ and $g(x,l)$, respectively. By Lemma 2.3.2(2), $\partial U[\bar{r},\bar{l}]$ does not contain x. By Lemma 4.1.5, s,t are on the far side of S from x. Hence by Corollary 2.4.2, $\bar{s},\bar{t} \in \partial U[\bar{r},\bar{l}]$, and in fact the counterclockwise order must be $\bar{r}$, $\bar{s}$, $\bar{t}$, $\bar{l}$, since $\partial U[\bar{s},\bar{t}]$ does not contain x. By Corollary 2.4.2, r, s, t, l appear in that counterclockwise order on C; since u appears between s and t, u is on the far side of S from x. Again by Corollary 2.4.2, $\bar{s}$, $\bar{u}$, $\bar{t}$ appear in that counterclockwise order on ∂U.

Now suppose, contrary to the Lemma, that $V(u)$ is not empty; we will obtain a contradiction. Since Voronoi cells are connected to ∂U, there is $y \in V(u) \cap \partial U$. Now $y \neq x$ since Voronoi cells are relatively open, and x lies on the boundary of $V(s)$ (and $V(t)$).

As s, u, and t are distinct extreme elements of S, the set $\{s,u,t\}$ is extreme. As $x \in \partial U$, it is impossible for x to lie in the interior of Δsut. Moreover, s, u, and t are on the far side of $\{s,u,t\}$ from x (in that counterclockwise order). Thus either $x \in g(s,t)$ or $\{s,u,t,x\}$ is extreme (with s, u, t, x, occurring in that counterclockwise order). In either case $g(x,u)$ intersects $g(s,t)$ and enters $U[s,t]$ at some point u' (as u is an extreme point of C lying counterclockwise between s and t). Assume $y \in U[x,u]$, the case $y \in U[u,x]$ is similar. Lemma 4.1.6 implies that $g(y,t)$ does not intersect $g(x,u)$. As t lies in the relative interior of $U[u,x]$ and y lies in $U[x,u]$, $g(y,t)$ must intersect $u\bar{u}$ at some point $u'' \neq u$. This implies that the portion of $\bar{g}(x,u)$ from u' to u'' inclusive is contained in the triangle Δyst; in particular $u \in \Delta yst$. By Lemma 2.2.1, this contradicts the choice of $u \neq s,t$ as a site furthest from y. □

Corollary 4.1.8 (Ordering Lemma) *The ordering of sites with nonempty Voronoi cells around ∂C is the same as the ordering of these Voronoi cells around ∂U.*

Lemma 4.1.9 *Suppose $\{u_1, ..., u_m\}$ and $\{v_1, ..., v_m\}$ are separated by some boundary geodesic $g(x, y)$. Then there is a total of at most $O(m + n)$ distinct links in paths $g(u_i, v_i)$, $i = 1, ..., m$.*

Proof: By a proof essentially identical to that of Lemma 4 of Suri [Sur87], for each i there are at most three links of $g(u_i, v_i)$ which are not links of the shortest path trees $T(x)$ or $T(y)$. (Three links are needed rather than one, as in [Sur87], because u, v need not be corners of ∂U.) □

Lemma 4.1.10 *For a given set S of sites, there are three boundary geodesics g_1, g_2, g_3 so that for any point $a \in \partial U$ and any site b furthest from a one of the geodesics separates a from b.*

Proof: Refer to Figure 4.3. Pick $x \in S$ arbitrarily. Let y be a site furthest from x and z a site furthest from y. We argue the case that x, y, z are distinct and appear in that counterclockwise order around ∂C; the case that $x = z$ is easier and the case that the order is x, z, y is similar.

We claim that we can choose a site w furthest from z in $\partial C[x, y]$. We first show that w can be chosen to lie in $\partial C[x, z]$: if we can only choose a w furthest from z in $\partial C[z, x] - \{x\}$, then in particular $d(z, w) > d(z, y)$. We also have $d(x, y) \geq d(x, z)$ and $d(y, z) \geq d(y, w)$, so adding all three inequalities we get $d(x, y) + d(z, w) > d(x, z) + d(y, w)$. This contradicts the triangle inequality as can be seen by considering a point in the intersection of $g(y, w)$ and $g(x, z)$. Now w cannot be in $\partial C[y, z] - \{y\}$, else $g(x, w)$ intersects $g(y, z)$, a contradiction of Lemma 4.1.6, taking $u = x$, $t = w$, $v = z$, and $s = y$. Hence we can choose w in $\partial C[x, y]$.

Let x^*, y^*, z^* and $\bar{y}, \bar{z}, \bar{w}$ be the closest foreshadows and closest shadows of $g(x, y)$, $g(y, z)$, and $g(z, w)$, respectively. We claim $y^* \in \partial U[x^*, \bar{y}]$: this is immediate if $y \in \partial U$ or $g(x, y)$ and $g(z, y)$ share a common final link; otherwise it follows from

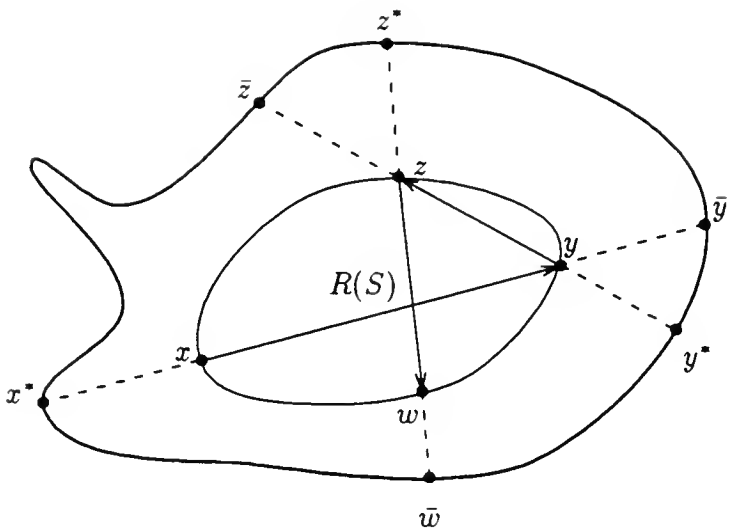


Figure 4.3: Illustration to the proof of Lemma 4.1.10

Lemma 2.2.4(1), since y^* is the closest shadow of $g(z,y)$. Similarly $z^* \in \partial U[y^*, \bar{z}]$ and $x^* \in \partial U[z^*, \bar{w}]$.

Now by the Extension Lemma, x^*, y^*, z^* have y, z, w as their respective furthest sites. By the Ordering Lemma, all points in $\partial U[x^*, y^*]$ have furthest sites in $\partial C[y,z]$; clearly $\bar{g}(x,y)$ separates $\partial U[x^*, y^*]$ from $\partial C[y,z]$. Similarly the sites furthest from $\partial U[y^*, z^*]$ lie in $\partial C[z,w]$ and these two sets are separated by $\bar{g}(y,z)$. Finally, the sites furthest from $\partial U[z^*, x^*]$ lie in $\partial C[w,y]$ and these two sets are separated by $\bar{g}(z,x)$. Thus $\bar{g}(x,y)$, $\bar{g}(y,z)$, and $\bar{g}(z,w)$ are three geodesics with the desired properties. Note that in fact x^* can have furthest sites both in $\partial C[y,z]$ and $\partial C[w,y]$, and similarly for y^* and z^*, while each of the remaining points of ∂U has its furthest sites in exactly one of $\partial C[y,z]$, $\partial C[z,w]$, $\partial C[w,y]$. □

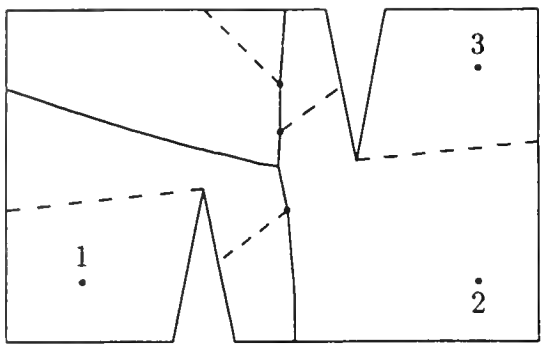


Figure 4.4: Refined Voronoi diagram of the sites of Figure 4.1.

4.1.3 The Refined Voronoi Diagram

The Voronoi diagram $\mathcal{V}$ clearly has $O(k)$ edges, since together with ∂U it forms a planar graph with at most k bounded regions, and all vertices of degree three or more. However, this is not an accurate description of the size complexity of the Voronoi diagram, since each Voronoi edge may consist of sections of several different hyperbolic arcs (as in the closest-site case). This Section discusses a refinement of the Voronoi partition, obtained by further subdividing each Voronoi cell $V(s)$ by the shortest path partition from s. Each bounding edge of a refined Voronoi cell is a line segment or a section of a hyperbola. The main theorem is a linear bound on the size complexity of the refined Voronoi diagram. This implies an $O(n+k)$ bound on the size complexity of the Voronoi diagram itself.

The *refined Voronoi cell of site s with anchor a*, $V_a^*(s)$, is $V(s) \cap P_a(s)$. The *refined bisector edge* $e_{ab}^*(s,t)$ is $e(s,t) \cap P_a(s) \cap P_b(t)$. The *refined partition edge (from s with anchor a)*, $p_a^*(s)$, is $V(s) \cap p_a(s)$. A *refined Voronoi edge* is a refined bisector edge or a refined partition edge. Refer to Figure 4.4. Observe that distinct refined Voronoi edges are disjoint (except possibly at their endpoints). Refined Voronoi edges consisting of a single point and empty refined Voronoi cells and edges are disregarded.

Suppose $e_{ab}^*(s,t)$ is not empty. It is easy to see that each endpoint of $e_{ab}^*(s,t)$ is

either a vertex of $\mathcal{V}$, a hitpoint, or a breakpoint (refer to Section 2.6). Moreover, $e^*_{ab}(s,t)$ does not contain breakpoints (except possibly as endpoints). Consequently $e^*_{ab}(s,t)$ is a hyperbolic arc or a line segment.

Lemma 4.1.11 *Either $p^*_a(s)$ is empty, or it is all of $p_a(s)$, or it has an open endpoint at a breakpoint of $e(s,t)$ for some site t and closed endpoint on ∂U. (The latter two cases are illustrated in Figure 4.4.)*

Proof: Suppose $p^*_a(s)$ is not empty and is not all of $p_a(s) = ay$ (where $y \in \partial U$). Then $p_a(s)$ must intersect some edge $e(s,t)$ before first entering $V(s)$. By the Extension Lemma, the intersection is a single point x, and $p^*_a(s) = xy - \{x\}$ is contained in $V(s)$. □

Suppose $p^*_a(s)$ is a refined partition edge of $V(s)$. Let $t(p^*_a(s))$ be the first link of the geodesic $g(a,s)$, directed towards s. Then by definition of the refined partition edge $p^*_a(s)$, $p_a(s)$ and $t(p^*_a(s))$ are collinear and meet at a. We claim that if $p^*_a(s)$ and $p^*_{a'}(s')$ are distinct refined partition edges, then $t(p^*_a(s))$ and $t(p^*_{a'}(s'))$ are distinct, at least as directed links. This claim certainly holds if $a \neq a'$. If $a = a'$, then we must have $s \neq s'$ by the previous Lemma. Since $p^*_a(s) \subseteq V(s)$ and $p^*_{a'}(s') \subseteq V(s')$, we must have $\theta(a,s) \neq \theta(a,s')$. Since $t(p^*_a(s))$ and $t(p^*_a(s'))$ have these respective directions, they must be distinct.

Lemma 4.1.12 *Link $t(p^*_a(s))$ is a link of $g(v,s)$, where v can be chosen to be a corner lying in $\partial U \cap V(s)$ or a hitpoint of $V(s)$ (i.e. an endpoint of $\partial U \cap V(s)$).*

Proof: Let $p_a(s) = ay$, $y \in \partial U \cap V(s)$. Segment ay partitions U into two polygonal regions U_1 and U_2 so that U_1 contains s and every geodesic to s from a point in U_2 contains $t(p^*_a(s))$. If we traverse $\partial U \cap U_2$ starting at y, we must encounter either the endpoint of $\partial U \cap V(s)$ or a corner of ∂U lying in $V(s)$. □

The *refined Voronoi diagram*, $\mathcal{V}^*$, is the union of all refined Voronoi edges. A *vertex* of $\mathcal{V}^*$ is a vertex of $\mathcal{V}$ or a breakpoint. A *hitpoint* of $\mathcal{V}^*$ is a hitpoint of $\mathcal{V}$ or the point of intersection of a refined partition edge with ∂U (in the case when the refined partition

edge coincides with a shortest path partition edge, only its non-anchor endpoint is considered a hitpoint).

Lemma 4.1.13 *There are at most $O(n+k)$ refined Voronoi edges and vertices.*

Proof: Clearly $\mathcal{V}$ has k cells, hence k hitpoints and $O(k)$ edges. We show that there are only $O(n+k)$ refined partition edges. Since each refined partition edge contributes a single breakpoint, it follows that there are only $O(n+k)$ refined bisector edges. By planarity, there are only $O(n+k)$ vertices in $\mathcal{V}^*$ as well.

By Lemma 4.1.10, there are three boundary geodesics g_1, g_2, g_3 and a partition of ∂U into three fragments σ_1, σ_2, σ_3 so that for $i = 1, 2, 3$, g_i separates every point in σ_i from its furthest sites. (This is not strictly true for the endpoints of σ_i, but the argument is similar.) Let W_i be the union of the sets of links of geodesics $g(v, w)$, where v is a corner of ∂U lying in σ_i and w is the unique site furthest from v or where v is a hitpoint of $\mathcal{V}$ lying in σ_i and w is one of the two sites whose Voronoi cell boundary contains v. Since there are k hitpoints and n corners in all of ∂U, there are certainly at most as many in each σ_i. By Lemma 4.1.9, there are $O(n+k)$ links in W_i. Now modify W_i so that it contains for each link two oppositely directed links; this doubles its size. By Lemma 4.1.12, if p_a^* is a refined partition edge, then $t(p_a^*)$ is in W_i, for some $i = 1, 2, 3$. Since each $t(p_a^*)$ corresponds to a *unique* refined partition edge, there are at most $O(n+k)$ such edges. □

4.1.4 Directing Edges of the Refined Voronoi Diagram

We let c be the (geodesic) center of C.

Necessarily c lies on some Voronoi edge, since there must be (at least) two sites attaining the maximum distance from c.

Lemma 4.1.14 *Suppose s is a site furthest from $x \in U$. Then the angle between $\theta(x, s)$ and $\theta(x, c)$ is less than $\pi/2$.*

Proof: Suppose the angle between $\theta(x,s)$ and $\theta(x,c)$ is at least $\pi/2$. By Lemma 2.2.2. $d(c,s) > d(x,s)$. But since s is a furthest site from x, $rad(x,C) = d(x,s)$. This contradicts the choice of c as the center of C. □

We use this Lemma to direct each refined Voronoi edge towards c (if there is a Voronoi edge containing c in its interior, we split the edge at c). We show how to direct an edge $e(s,t)$ of the (unrefined) Voronoi diagram; this direction extends to each refined bisector edge. A similar argument directs each refined partition edge. Suppose $x \in e(s,t)$, $x \neq c$. Since $e(s,t)$ bisects $\angle sxt$ and both the angle between $\theta(x,c)$ and $\theta(x,s)$ and the angle between $\theta(x,c)$ and $\theta(x,t)$ are less than $\pi/2$, it is not possible that $\theta(x,c)$ is perpendicular to $e(s,t)$ at x. Hence we can direct $e(s,t)$ towards c locally at x. If $c \notin e(s,t)$, this direction extends globally to $e(s,t)$ (since direction towards c is a continuous function away from the corners and can never be perpendicular to $e(s,t)$). If $c \in e(s,t)$, then $e(s,t)$ is split at c into two portions, each of which is consistently directed towards c.

Lemma 4.1.15 *Suppose $v \notin \partial U$ is a vertex of $\mathcal{V}^*$. Then there are at least two edges of $\mathcal{V}^*$ entering v and exactly one edge of $\mathcal{V}^*$ leaving v.*

Proof: We consider the case that v is in fact a vertex of $\mathcal{V}$; the case that v is a breakpoint is similar. For simplicity assume there are exactly three sites equidistant from v. Label the sites r,s,t so that $\theta(v,r)$, $\theta(v,s)$, and $\theta(v,t)$ are in counterclockwise order leaving v so that $m\angle rvt < \pi$. (This is possible by the previous Lemma as each of the angles formed between $\theta(v,c)$ and $\theta(v,r)$, $\theta(v,s)$ or $\theta(v,t)$ has measure less than $\pi/2$.) By Lemma 4.1.2, Voronoi edges $e(r,s)$, $e(s,t)$, and $e(t,r)$ are incident to v and extend in directions bisecting angles $\angle svr$, $\angle tvs$, and $\angle rvt$, respectively. Hence edges $e(r,s)$ and $e(s,t)$ enter v and $e(t,r)$ leaves v. The general case of a vertex of arbitrary high degree is handled analogously. Vertices $v \notin \partial U$ of degree two or less do not occur by definition of $\mathcal{V}^*$. □

Suppose Voronoi edge $e(s,t)$ intersects ∂U at hitpoint x; recall that by the general position assumption x is not a corner of ∂U. Then $e(s,t)$ is directed into the interior

of U, because $\theta(x,c)$ makes an angle strictly less than $\pi/2$ with both $\theta(x,s)$ and $\theta(x,t)$, and none of $\theta(x,s)$, $\theta(x,t)$ and $\theta(x,c)$ can leave U at x. A similar argument shows that refined partition edges are directed away from ∂U at hitpoints.

Corollary 4.1.16 *The unrefined Voronoi diagram $\mathcal{V}$ forms a (directed) tree with root c and edges directed towards c.*

Proof: No cycles are possible because otherwise some Voronoi cell would be separated from ∂U. Only c has out-degree zero; every other vertex or hitpoint of $\mathcal{V}$ has out-degree 1, so $\mathcal{V}$ is a root-directed tree. □

The refined Voronoi diagram $\mathcal{V}^*$ consists of $\mathcal{V}$ together with refined partition edges. Each such edge must lie entirely within a single Voronoi cell, having one endpoint on ∂U and the other endpoint at a reflex corner or on $\mathcal{V}$ (by Lemma 4.1.11).

4.2 The Algorithm

This Section contains the algorithm for computing $\mathcal{V}^*$. An outline of the algorithm is given in Figure 4.5. Step 1 can be performed in time $O(n \log n)$ [GJPT78]. The relative convex hull computation of the second step can be accomplished in time $O((n+k)\log(n+k))$ [Tou86]. The third step also takes time $O((n+k)\log(n+k))$ and is described in Lemma 4.2.1 below. The fourth step, the most difficult of the algorithm, is the computation of $\mathcal{V}^*$ restricted to ∂U. It is discussed in Sections 4.2.1 through 4.2.4. The last step is the extension of $\mathcal{V}^*$ to the interior of U. This is done using a "reverse geodesic sweeping" algorithm, discussed in Section 4.2.5 below. Both the fourth and the fifth steps take time $O((n+k)\log(n+k))$.

The computation of $\mathcal{V}^*$ restricted to ∂U is quite similar in outline to Suri's algorithm for furthest geodesic neighbors inside a simple polygon[Sur87]. We first reduce the computation of $\mathcal{V}^*$ on ∂U to at most three instances of the "two-fragment problem." Roughly, an instance of the two-fragment problem consists of a fragment of ∂U and a fragment of ∂C so that all furthest sites of points in the fragment of ∂U

Input: A polygon U with n sides and a set $S \subseteq U$ of k sites.
Output: The refined furthest-site geodesic Voronoi diagram $\mathcal{V}^* = \mathcal{V}^*_U(S)$.

1. Triangulate U.
2. Compute C, the relative convex hull of S, and discard all non-extreme sites of S.
3. Determine two or three two-fragment instances so that the union of the source fragments is ∂U.
4. Compute $\mathcal{V}^* \cap \partial U$ by calling $rgfs(u, v, s, t)$ for each two-fragment instance (u, v, s, t).
5. Call *sweep* to extend $\mathcal{V}^*$ to the interior of U.

Figure 4.5: Procedure *gfv*.

are contained in the fragment of ∂C (the objective is computing $\mathcal{V}^*$ restricted to the fragment of ∂U). Such a pair must also satisfy a technical condition given below: this reduction appears in Section 4.2.1. The algorithm to solve the two-fragment problem is based on a divide-and-conquer schema that splits an instance into two smaller instances. The basic properties of the divide-and-conquer schema appear in Section 4.2.2. Section 4.2.3 contains the exact splitting method and the procedures for handling the base cases of the recursion. The complexity analysis appears in Section 4.2.4. We show that the sum of all instance sizes at each level of recursion is linear in $n + k$. This implies over all $O((n + k) \log(n + k))$ running time.

We work with polygonal relatively convex sets in addition to simple polygons. Any such relatively convex set Q can be decomposed into a collection of plateaus and ridges. Clearly a triangulation of Q can be obtained just by triangulating each plateau in the decomposition. This is easily done in time $O(m \log m)$ [GJPT78], if m is the number of segments in the boundary of Q. Similarly a shortest path partition of Q from an arbitrary point in it can be obtained by using a shortest path partition

algorithm in each plateau of the decomposition. If Q has been triangulated, this takes time $O(m)$ [GHL*87].

4.2.1 The Two-Fragment Problem

A *two-fragment instance* is a quadruple (u, v, s, t) where $u, v \in \partial U$, $u \neq v$, s is a site furthest from u, t is a site furthest from v (possibly $s = t$), and $g(u, v)$ separates $\partial U[u, v]$ from $\partial C[s, t]$. The *two-fragment problem* is "Given two-fragment instance (u, v, s, t), compute $\mathcal{V}^* \cap \partial U[u, v]$." Observe that by the Ordering Lemma, only the Voronoi cells of sites in $\partial C[s, t]$ can intersect $\partial U[u, v]$. The *source fragment* of two-fragment instance (u, v, s, t) is $\partial U[u, v]$; the *target fragment* is $\partial C[s, t]$.

Lemma 4.2.1 *There exists a set of at most three instances of the two-fragment problem so that the union of the source fragments is ∂U. The instances each have size $O(n + k)$ and can be computed in time $O((n + k)\log(n + k))$ given a triangulation of U.*

Proof: Choose x, y, z, w, x^*, y^*, z^* as in the proof of Lemma 4.1.10. It is clear that (x^*, y^*, y, z), (y^*, z^*, z, w), and (z^*, x^*, w, y) are two-fragment instances each of size at most $O(n+k)$; their source fragments cover ∂U. As for computing them, the choice of x was arbitrary. Site y can be determined in time $O((n+k)\log(n+k))$ by computing the shortest path tree from x, then determining the distance from every site to x using a planar point location algorithm in the resulting shortest path partition. Sites z and w can be determined similarly. Surely x^*, y^*, z^* can be computed in additional time $O(n)$. The case when $x = z$ is handled similarly. □

Let D be the relative convex hull of the sites on $\partial C[s, t]$. Clearly the ordering of sites on D is the same as on C, with the addition that s immediately follows t in counterclockwise order. Let $R(u, v, s, t)$ be the relative convex hull of $\partial U[u, v]$ and D. See Figure 4.6.

We say (u, v, s, t) is *degenerate* if D is contained in $g(u, v)$. If (u, v, s, t) is degenerate, there can be no sites in D besides s and t and, by Lemma 4.1.6, the order along

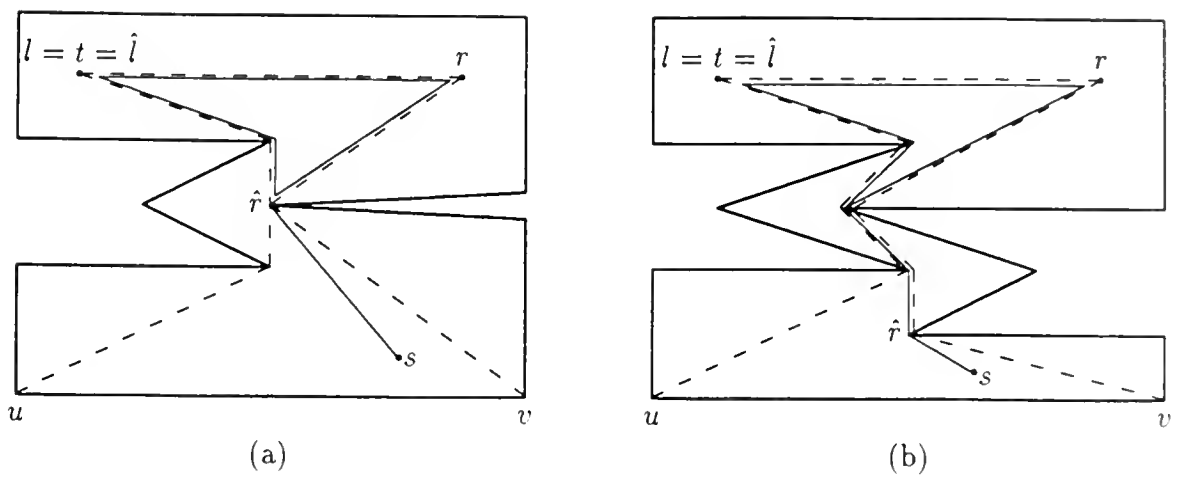


Figure 4.6: Solid outlines D; dashes outline $R(u, v, s, t)$.

$g(u, v)$ must be u, t, s, v. In this case we define $l = t$ and $r = s$.

Suppose (u, v, s, t) is not degenerate. Let l be the counterclockwise extreme point of D from u and r the clockwise extreme point of D from v. By Lemma 2.4.3, $\partial U[u, v]$, $g(v, r)$, $\partial D[r, l]$ (if $r \neq l$), and $g(l, u)$ constitute a counterclockwise traversal of the boundary of $R(u, v, s, t)$. It is an immediate consequence of the following Lemma that $\partial D[r, l]$ is a subpath of $\partial C[s, t]$ (if $r = l$, we only require $r = l \in \partial C[s, t]$).

Lemma 4.2.2 *Sites s, r, l, t are in that counterclockwise order on ∂D, not necessarily all distinct.*

Proof: The Lemma is trivial if $s = t$ or if (u, v, s, t) is degenerate, so assume $s \neq t$ and (u, v, s, t) is not degenerate. Assume $r \neq l$. We show that if s appears on $\partial D[r, l]$ then $r = s$ and if t appears on $\partial D[r, l]$, then $t = l$. Since s is the extreme point of D immediately counterclockwise from t, this implies the Lemma.

Suppose t appears on $\partial D[r, l]$ and $t \neq l$. Since s is immediately counterclockwise from t, s is also an extreme point of D lying on $\partial D[r, l]$ and thus an extreme point of $R(u, v, s, t)$. Then t, s, u, and v appear in that order on a counterclockwise traversal

of $\partial R(u,v,s,t)$, which by Lemma 2.3.7 implies that $g(t,u)$ meets $g(v,s)$, contrary to Lemma 4.1.6.

Turning to the case $r = l$, the Lemma is contradicted only if s, t, $r = l$ are distinct and lie in that counterclockwise order on ∂D. But this is impossible, as s immediately follows t on ∂D. □

We wish to give a definition of "left side connector" and "right side connector" to capture the bounding edges of $R(u,v,s,t)$ not in $\partial C[r,l]$ and $\partial U[u,v]$. The obvious definitions are $g(u,l)$ and $g(v,r)$, respectively. Unfortunately, these definitions are not adequate. In Section 4.2.4, we analyze the size of side connectors; one crucial property used in our argument is that, if $s \neq t$, then the left and right side connectors are disjoint except possibly at their endpoints (Lemma 4.2.3 below). Unfortunately, this is not true for side connectors defined as $g(u,l)$ and $g(v,r)$. See Figure 4.6(b).

It is clear that geodesic $g(v,r)$ has connected intersection with ∂D; furthermore if the intersection is more than a point it must be some final portion of geodesic $g(r',r)$, r' the site of D immediately clockwise of r. Let $\hat{r}$ be r if $r = s$, otherwise let $\hat{r}$ be the first point of $g(r',r)$ intersected by $g(v,r)$. The *right side connector* of (u,v,s,t) is $g(v,\hat{r})$. Similarly, we define the *left side connector* of (u,v,s,t) to be $g(u,\hat{l})$ where $\hat{l}$ is l if $l = t$, otherwise $\hat{l}$ is the first point of $g(l',l)$ intersected by $g(u,l)$, where l' is the site of D immediately counterclockwise from l. See Figure 4.6. A *connector edge* is a link in either the left or right connector.

The *size* of (u,v,s,t), denoted $\|(u,v,s,t)\|$, is $\|\partial C[s,t]\| + \|\partial U[u,v]\|$ plus the sizes of the side connectors. Since $\partial D[r,l]$ is a subpath of $\partial C[s,t]$, it is clear that $\|\partial R(u,v,s,t)\| \leq \|(u,v,s,t)\|$.

Lemma 4.2.3 *If the side connectors of a two-fragment instance (u,v,s,t) meet at a point other than one of their endpoints, then $s = t$.*

Proof: If (u,v,s,t) is degenerate, then u, $l = t$, $r = s$, v appear in that order along $g(u,v)$, and the side connectors are disjoint unless $l = r$, which in turn forces $s = t$. So suppose (u,v,s,t) is not degenerate and point $a \neq u,v,\hat{r},\hat{l}$ is common to both

side connectors. Without loss of generality we may assume that a is a reflex corner of ∂U. Deleting a splits $R(u,v,s,t)$ into two components, whose closures R_1 and R_2 are relatively convex polygonal regions with the property that a geodesic connecting two points of $R(u,v,s,t) - \{a\}$ passes through a if and only if one of the points lies in $R_1 - \{a\}$ and the other in $R_2 - \{a\}$. Since a lies on the left connector and thus on $g(u,l)$, either $u \in R_1$ and $l \in R_2$ or vice versa (and similarly for v and r). Since $\partial U[u,v]$, $g(v,r)$, $\partial C[r,l]$, $g(l,u)$ constitute a traversal of $\partial R(u,v,s,t)$ and such a traversal cannot visit any point more than twice, $\partial U[u,v]$ and $\partial C[r,l]$ do not meet a. In particular, u and v lie in the same component. Similarly, r and l must lie in the same component.

Without loss of generality, assume that R_1 contains u and v and R_2 contains r and l. Now also $s \notin R_1$, else a would appear on $\partial D[s,r]$ and thus could not lie on a side connector. Hence $s \in R_2$, similarly $t \in R_2$. Thus s is furthest from u, t is furthest from v, but both $g(u,t)$ and $g(v,s)$ pass through a, implying $s = t$ by Lemma 4.1.6. □

4.2.2 The Recursion Scheme

Figure 4.7 contains a recursive procedure *rgfs* for solving the two-fragment problem. Section 4.2.3 discusses the base cases and the choice of the splitting point w, while the complexity analysis is contained in Section 4.2.4. We now discuss some basic data structures needed for the recursion.

At each level of recursion, we need to have available the boundary of $R(u,v,s,t)$ and a triangulation of its interior. For the topmost level, the boundary of $R(u,v,s,t)$ can be constructed using the relative convex hull algorithm of Toussaint[Tou86]; this takes time $O((n+k)\log(n+k))$. Then it can be triangulated in additional time $O((n+k)\log(n+k))$ [GJPT78].

For the recursive step, having determined the splitting point w, we need to compute $f(w)$ and the boundaries and triangulations of both $R(u,w,s,f(w))$ and $R(w,v,f(w),t)$ in total time $O(\|(u,v,s,t)\|)$. To compute $f(w)$ it suffices to know

```
PROCEDURE rgfs(u, v, s, t)
    IF ∂U[u, v] or ∂C[s, t] is a base case
        THEN compute ∂U[u, v] ∩ V* directly
        ELSE choose w ∈ ∂U[u, v]
             locate a site f(w) furthest from w
             call rgfs(u, w, s, f(w)) and rgfs(w, v, f(w), t)
END
```

Figure 4.7: Recursive procedure *rgfs*.

the geodesic distance $d_w(r)$ for every site r in $\partial C[s,t]$; $d_w(r)$ can be determined in constant time if the cell of the shortest path partition of $R(u,v,s,t)$ from w containing r is known. The cell containing r for all sites r in $\partial C[s,t]$ can be determined in total time $O(\|(u,v,s,t)\|)$ as follows. We assume the shortest path partition of $R(u,v,s,t)$ from w has been computed and refined to a triangulation (this takes only linear additional time); hence each cell is a triangle. First locate the triangle containing s; this clearly can be done in the allowed time bound. Then traverse $\partial C[s,t]$, in one step moving to the next vertex of $\partial C[s,t]$ or to the next intersection of the current edge of $\partial C[s,t]$ with the boundary $\partial\Delta$ of the current triangle Δ of the shortest path partition from w. Notice that the intersection of ∂C with Δ has at most three connected components, since C is relatively convex. Hence the traversal of $\partial C[s,t]$ takes total time $O(\|(u,v,s,t)\|)$, as the charge for a step to a vertex of $\partial C[s,t]$ can be allotted to the vertex and the step to an intersection with Δ can be allotted to one of the three connected components of $\Delta \cap \partial C[s,t]$.

We compute the boundary and triangulation of $R(u,w,s,f(w))$ as follows (handling $R(w,v,f(w),t)$ is similar). If $(u,w,s,f(w))$ is degenerate, both the boundary and triangulation of $R(u,w,s,f(w))$ can be easily obtained from the shortest path partition of $R(u,v,s,t)$ from w. Otherwise compute r', the clockwise extreme point

of $\partial C[s, f(w)]$ from w using the shortest path partition from w. Similarly l', the counterclockwise extreme point of $\partial C[s, f(w)]$ from u, can be determined by computing the shortest path partition of $R(u, v, s, t)$ from u. Now since l' is extreme in $R(u, w, s, f(w))$, $g(w, l')$ splits $R(u, w, s, f(w))$ into two pieces, one piece lying to the left and one to the right. (Possibly one or the other is just $g(w, l')$.) A triangulation of the piece lying to the right can be obtained by refining the shortest path partition of $R(u, v, s, t)$ from w. Similarly a triangulation of the piece lying to the left can be obtained by refining the shortest path partition of $R(u, v, s, t)$ from l'. Notice the links in $g(u, l')$, $g(w, l')$ and $g(w, r')$ are used as triangle edges in this triangulation. The left and right side connectors of $R(u, w, s, f(w))$ are easily determined from $g(u, l')$ and $g(w, r')$. This computation can clearly be done in time $O(\|(u, v, s, t)\|)$.

4.2.3 Choosing Splitting Points and the Base Cases

If u and v do not lie on the same wall of ∂U, then splitting point w is chosen simply as the corner of $\partial U[u, v]$ so that $-1 \leq \|\partial U[u, w]\| - \|\partial U[w, v]\| \leq 1$.

If $\partial C[s, t]$ is a single site, i.e., $s = t$, then all of $\partial U[u, v]$ lies in $V(s)$. We need to find the refined partition edges of $\mathcal{V}^*$ intersecting $\partial U[u, v]$; it is sufficient to compute the shortest path partition of $R(u, v, s, t)$ from s, which can be done in time $O(\|(u, v, s, t)\|)$ given the triangulation of $R(u, v, s, t)$[GHL*87].

It is possible that $s \neq t$ but u and v lie on the same wall of ∂U. In this case there is no obvious splitting point w. We perform a "partition" step: segment uv is split into subsegments so that within each subsegment the shortest path tree from any point on the subsegment to the sites on $\partial C[s, t]$ is combinatorially invariant. This partitioning is described below; it results in $O(\|(u, v, s, t)\|)$ subsegments and takes time $O(\|(u, v, s, t)\| \log \|(u, v, s, t)\|)$. We introduce the partition points as dummy vertices, and use them as splitting points in the divide and conquer. Notice that the partitioning is performed at most once on a path from the topmost instance to a leaf instance in the recursion tree. Hence we introduce only $O(n + k)$ such points, since the sum of instance sizes at each level of the recursion tree is $O(n + k)$ (as

will be shown in Section 4.2.4). By the same token, the total time to compute the partitioning points is $O((n+k)\log(n+k))$.

The remaining problem is to handle a base case instance (u, v, s, t), where $s \neq t$ and the shortest path tree is combinatorially invariant on segment $uv = \partial U[u, v]$. Because of this invariance, no refined partition edge of $\mathcal{V}^*$ intersects segment uv, in other words $uv \cap \mathcal{V}^* = uv \cap \mathcal{V}$ is the set of bisector hitpoints on uv. Notice that, the partition induced on segment uv by $\mathcal{V}^*$ is exactly the partition induced by the upper envelope of the functions d_r, where r is a site in $\partial C[s, t]$. Again because of the combinatorial invariance of the shortest path tree, each function d_r is "simple" on segment uv; specifically $d_r(x)$ is of the form $c_1 + \sqrt{c_2(x)}$ where c_1 is constant and $c_2(x)$ depends quadratically upon the position of x on segment uv. (Observe that the purpose of partitioning the original wall was to ensure that c_1 and c_2 are fixed over the length of uv. Their values for each site $r \in \partial C[s, t]$ are defined by the identity of the anchor of $x \in uv$ with respect to r and the distance from r to this anchor; all of this information can be determined in linear time from the shortest-path tree of $R(u, v, s, t)$ from, say, u.) Thus in constant time it is possible to determine, for a pair of sites r and r' the (unique) point $x \in uv$, if any, for which $d_r(x) = d_{r'}(x)$. Now by the Ordering Lemma, Voronoi cells appear along segment uv in the same order as the corresponding sites appear along $\partial C[s, t]$. This implies that the upper envelope of the functions d_r on segment uv can be computed in time proportional to the number of sites (which is certainly $O(\|(u, v, s, t)\|)$). For example, an incremental algorithm is sufficient. Suppose that the partition of segment uv induced by an initial subsequence of the sites on $\partial C[s, t]$ has been computed. Then the partition induced by adding the next site in order can be determined in constant time plus time proportional to the number of cells deleted from the partition of segment uv computed so far.

We now describe the partition step, which uses a technique similar to that of [GHL*87]. We actually partition segment uv so that within each subsegment, the shortest path tree to every site of $\partial C[s, t]$ and every vertex of $\partial R(u, v, s, t)$ is combinatorially invariant. To do this, compute $T(u)$, the shortest path tree of $R(u, v, s, t)$

from u. Include in $T(u)$ the links of geodesics from u to sites of $\partial C[s,t]$ not appearing on $\partial R(u,v,s,t)$ (there can be at most one such link per site not already in $T(u)$). The augmented $T(u)$ can be computed in total time $O(\|(u,v,s,t)\|)$ using the technique described for computation of $f(w)$ in Section 4.2.2. Similarly compute $T(v)$, the augmented shortest path tree from v.

For each z, z either a site of $\partial C[s,t]$ or a vertex of $\partial R(u,v,s,t)$, compare $\theta(z,u)$ with $\theta(z,v)$. This is possible using $T(u)$ and $T(v)$. If $\theta(z,u) = \theta(z,v)$, do nothing. If z lies on segment uv, do nothing. Otherwise geodesic triangle Δzuv must actually be a simple polygon (since uv is a wall). Pass a line through the first link of $g(z,u)$; it must hit segment uv since the interior angles of Δzuv are reflex except at z,u,v. The intersection point is a partition point. Similarly obtain a partition point from the first link of $g(z,v)$. It is easy to see that the partition points generated in this fashion form the required partition of segment uv. Generating the partition points takes $O(\|(u,v,s,t)\|)$ time and sorting them requires $O(\|(u,v,s,t)\| \log \|(u,v,s,t)\|)$ time.

4.2.4 Complexity Analysis

Lemma 4.2.4 *Let (u_i,v_i,s_i,t_i) be a topmost two-fragment instance. There are at most $O(n+k)$ distinct connector edges among all subinstances of (u_i,v_i,s_i,t_i).*

Proof: By the discussion of the preceding Section, there are, over the course of algorithm execution, only $O(n+k)$ subinstances of the two-fragment problem, hence as many connectors. Since $g(u_i,v_i)$ separates $\partial U[u_i,v_i]$ from $\partial C[s_i,t_i]$, it separates the endpoints of each connector as well. Hence by Lemma 4.1.9, there are at most $O(n+k)$ distinct connector edges. □

Lemma 4.2.5 *Let (u_i,v_i,s_i,t_i) be a topmost two-fragment instance. At each level of recursion, each connector edge appears in only a constant number of subinstances of (u_i,v_i,s_i,t_i).*

Proof: We count the number of times edge ab can appear as a left connector edge directed from a to b. Notice a and b may be assumed to be reflex corners of ∂U, for otherwise ab is necessarily a first or last link on a side connector, and hence can appear in only one left connector (three if ab is the last link and the target fragment consists of the single site b in two of the instances). Let Q_a be the set of points x of U for which $g(x,b)$ passes through a. Similarly, define Q_b as the set of points that can reach a only through b. Clearly Q_a and Q_b are non-empty and disjoint. Let A be $\partial U[u_i,v_i] \cap Q_a$ and B be $\partial C[s_i,t_i] \cap Q_b$. It can be checked that A is a single fragment of ∂U and that B is a single fragment of $\partial C \cap S$. Then for $u \in \partial U[u_i,v_i]$ and $t \in \partial C[s_i,t_i]$, link ab appears in $g(u,t)$ exactly if $u \in A$ and $t \in B$.

We first claim that at each level of recursion there are at most two nonleaf instances of the two-fragment problem for which both the source fragment intersects A and the target fragment intersects B. To see this suppose (u_j,v_j,s_j,t_j), $j = 1,2,3$, are instances at the same level, $\partial U[u_j,v_j] \cap A \neq \emptyset$, $\partial C[s_j,t_j] \cap B \neq \emptyset$, and u_1,u_2,u_3 (and hence s_1,s_2,s_3) are in that counterclockwise order. Since these instances are all at the same level of recursion, v_2 appears between u_2 and u_3 (possibly $v_2 = u_3$) and s_2 appears between t_1 and t_2 (possibly $s_2 = t_1$). But then $v_2 \in A$, $s_2 \in B$, so the geodesic $g(v_2,s_2)$ contains link ab. Recalling that the relative interior of ab does not lie in Q_b and hence is disjoint from $\partial C[s_2,t_2] \subseteq B$, we deduce that the right connector of (u_2,v_2,s_2,t_2) contains the relative interior of ab. Since $g(u_2,t_2)$ also contains ab, the relative interior of ab lies in the left connector of this instance as well. Hence $s_2 = t_2$ by Lemma 4.2.3 and (u_2,v_2,s_2,t_2) is a leaf instance.

We now claim that at each level of recursion there are at most four instances of the two-fragment problem (leaf and nonleaf) with ab appearing as a left connector edge. Such an instance (u,v,s,t) must have $u \in A$ and $t \in B$. This is only possible if, for its parent instance, the source fragment intersects A and the target fragment intersects B. As just argued there are only two such parent instances. □

Theorem 4.2.6 *$\mathcal{V}^* \cap \partial U$ can be computed in time $O((n+k)\log(n+k))$.*

Proof: Clearly U can be triangulated in time $O(n \log n)$ [GJPT78]. By Lemma 4.2.1,

there are three two-fragment instances with union of source fragments equal to ∂U that can be computed in time $O(n+k)$. We show that *rgfs* solves the two-fragment problem in total time $O((n+k)\log(n+k))$ for each (top-level) instance, proving the Theorem.

Consider the work performed by *rgfs* for all instances at a particular level of recursion, ignoring recursive calls and the time required to partition walls as discussed in Section 4.2.3. It is linear in instance size which is the sum of the sizes of source and target fragments and the sizes of the side connectors. The total size of source and target fragments at a particular level of recursion is $O(n+k)$, because source and target fragments are partitioned disjointly except for endpoints, and there are only $O(n+k)$ possible endpoints. By Lemmas 4.2.4 and 4.2.5, the total size of all connectors at a particular level of recursion is also $O(n+k)$. Hence the total work at a particular level of recursion, summed over all instances at the level, is $O(n+k)$.

The total depth of recursion is $O(\log(n+k))$: at each step except for partition steps, the size of a source fragment is split in half. At a partition step, the size of the source fragment increases to at most $O(n+k)$, and a partition step happens at most once on a path in the recursion tree from topmost instance to leaf instance.

The total work required for partitioning is $O((n+k)\log(n+k))$. Hence the total work to solve a single top-level two-fragment instance is $O((n+k)\log(n+k))$. □

4.2.5 Computing $\mathcal{V}^*$

$\mathcal{V}^*$ is computed by the procedure *sweep* (Figure 4.8), which is a "reverse geodesic sweeping algorithm"; it progresses from ∂U towards c, the center of C.

Theorem 4.2.7 *Procedure* sweep *computes* $\mathcal{V}^*$. *It can be implemented to run in time* $O((n+k)\log(n+k))$ *and space* $O(n+k)$.

Proof: For positive real z, let $D_c(z)$ be the geodesic disc of radius z centered at c, i.e., the set of all points of U at geodesic distance at most z from c. We claim that

Input: Triangulated polygon U, refined Voronoi diagram $\mathcal{V}^* = \mathcal{V}^*_U(S)$ restricted to ∂U.
Output: $\mathcal{V}^*$.

Data structures:
L: a doubly-linked circular list of refined Voronoi edges.
Q: a priority queue of points of U, ordered by decreasing geodesic distance from c.

```
PROCEDURE sweep
    Compute c, the center of C.
    Compute the shortest path partition of U from c.
    Initialize L to be ∂U ∩ V*.
    Initialize Q to contain all pairwise intersections of refined Voronoi edges immediately
    adjacent in L and all anchors of the refined partition edges appearing in L.
    WHILE Q ≠ ∅
        Extract from Q the point v of maximum geodesic distance from c.
        Delete from L all refined Voronoi edges with head v.
        Delete from Q any intersections or anchors involving the edges just deleted.
        IF v is not an anchor THEN
            Insert into L the refined bisector edge e with tail v.
            Insert into Q any new intersections of e with adjacent refined Voronoi edges.
        END IF
    END WHILE
END sweep
```

Figure 4.8: Procedure *sweep*.

the WHILE loop maintains the invariant that L contains exactly the refined Voronoi edges intersected by $\partial D_c(r)$, in order around $\partial D_c(r)$ where r is the distance from c to the most recently processed vertex of $\mathcal{V}^*$. This follows from Lemma 4.1.15, using standard sweepline arguments [BO79]. Hence procedure *sweep* computes $\mathcal{V}^*$.

List L can be implemented simply as a circular doubly-linked list, so each list operation takes constant time. Q can be implemented as a heap, so that each operation takes time $O(\log(n+k))$. The geodesic center c of C can be computed in time $O((n+k)\log(n+k))$ as follows. Pollack, Sharir, and Rote [PSR] show how to compute the center of (the set of vertices of) a simple polygon; their algorithm extends easily to a polygonal relatively convex set as well. Since geodesics restricted to be inside a relatively convex set (with respect to U) are identical to geodesics inside U, it suffices to compute the center of $C = R(S)$.

The shortest path partition of U from c can be computed in time $O(n)$ since U is triangulated [GHL*87]. Given the shortest path partition from c, the geodesic distance from a point $x \in U$ to c can be computed in time $O(\log n)$, using a planar subdivision search algorithm (such as that of [ST86]) to locate the shortest path partition cell containing x. By Theorem 4.2.6, $\partial U \cap \mathcal{V}^*$ can be computed in time $O((n+k)\log(n+k))$. Hence L and Q can be initialized in total time $O((n+k)\log(n+k))$.

Each iteration of the WHILE loop uses a constant number of operations on Q and L and one geodesic-distance computation for each item inserted in Q; hence each iteration takes time $O(\log(n+k))$. By Lemma 4.1.13, there are only $O(n+k)$ iterations of the WHILE loop. Thus the running time of the entire algorithm is $O((n+k)\log(n+k))$. Clearly the space usage is $O(n+k)$. □

4.3 Open Problems

It remains an open problem to determine whether the closest-site geodesic Voronoi diagram of a general set of k point sites in an n-gon can be computed in time $O((n+$

$k)\log(n+k))$ or, for that matter, any faster than $\Omega((n+k)\log(n+k)\log n)$. Let us observe that the Euclidean closest-pair problem is linear-time reducible to the closest-site geodesic Voronoi diagram problem as follows: Given a set S of k points, construct in linear time a rectangular box B large enough to enclose the convex hull of S. At this point, computing the closest-site diagram $\mathcal{V}_B(S)$ produces the (conventional closest-site) Voronoi diagram of S truncated to B. However, it has been shown (see, for example, [PS85]) that the shortest Euclidean distance between two points of S is realized by a pair of points whose Voronoi cells are adjacent and the segment connecting which intersects the edge common to the two cells–hence the intersection point must lie in the convex hull of S and thus in B. In particular, we can locate the closest pair in linear time given the truncated closest-site Voronoi diagram, which shows an $\Omega(k\log k)$ lower bound for computing the geodesic closest-site Voronoi diagram. Recalling the $\Theta(n+k)$ bound on the complexity of the diagram, we obtain the best currently known lower bound of $\Omega(n+k\log k)$. In particular, this implies that the extension scheme, as presented in Section 3.3.1, cannot be implemented in $O(n+k)$ time (assuming that the rest of the algorithm is not modified), as this would yield an $O((n+k)\log n)$ algorithm for constructing the geodesic closest-point Voronoi diagram, which is faster than the above lower bound when $k \gg n$.

A similar argument, employing the $\Omega(k\log k)$ lower bound for computing the diameter of a set of k points in the plane (see, for example, [PS85]), yields an $\Omega(n+k\log k)$ lower bound for computing the furthest-site geodesic Voronoi diagram.

There are a number of other open problems quite closely related to geodesic Voronoi diagrams. One of them is removing the general position assumptions and defining a cell of a *set* of sites to contain all points of U to which all sites of the set are simultaneously closest (see, for example, [ES86]). The objective, once again, would be to estimate the complexity of this diagram and devise an efficient algorithm for constructing it. Another possible avenue for extending our results is removing the restriction that sites be points. Can our approach be extended to yield an efficient algorithm for computing the diagram of a set of straight-line segments in a polygon

under a suitable metric, e.g., using an appropriate modification of Yap's technique [Yap87]? Observe that some of the properties we have discussed do not depend on the fact that sites are points—for example, Lemma 3.3.14 holds for any class of sites.

It might also be interesting to consider which properties of the geodesic distance (such as those discussed in Chapter 2 and Sections 3.1 and 3.2) survive if we allow the sides of U to be curvilinear.

A somewhat unrelated question that can be raised in conjunction with a geodesic closest-site Voronoi diagram is whether the sweeping approach of Fortune [For87] can be generalized for constructing such diagrams. A significant obstacle to such a generalization is the absence of an obvious natural equivalent of a straight sweep-line.

Part II

Entering the Third Dimension

Chapter 5

Triangles in Space, or Building (and Analyzing) Castles in the Air

5.1 Introduction

5.1.1 Terminology

Consider a collection G of n (possibly intersecting) closed flat triangular (or, more generally, arbitrary convex) objects $\Delta_1, ..., \Delta_n$ in three-dimensional Euclidean space $\mathbf{R}^3$ (see Figure 5.1). Let $A = A(G)$ denote the *arrangement* of these triangles, i.e., the subdivision of $\mathbf{R}^3$ induced by them; thus A is a decomposition of 3-space into pairwise disjoint connected cells of 0, 1, 2, or 3 dimensions, where

1. a 3-dimensional cell is a connected component of $\mathbf{R}^3 - \bigcup_i \Delta_i$,
2. a 2-dimensional cell (a *face*) is a connected component of $int(\Delta_i) - \bigcup_{j \neq i} \Delta_j$ for some i (where $int(\Delta)$ denotes the relative interior of Δ),

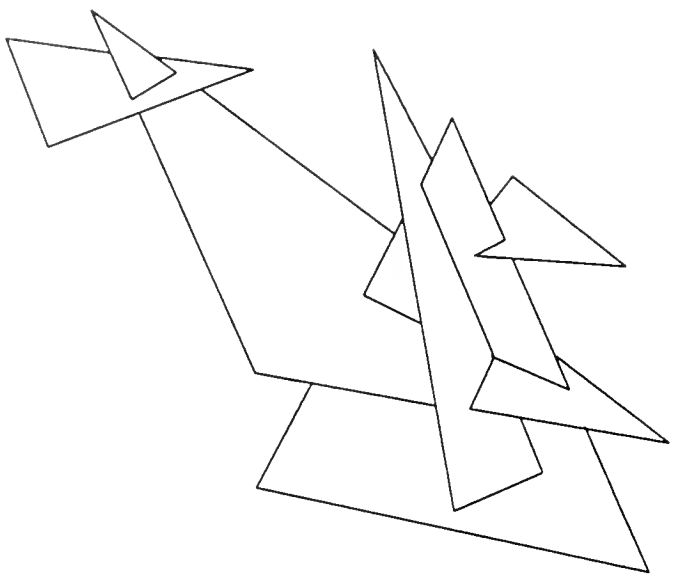


Figure 5.1: An arrangement of triangles in $\mathbf{R}^3$.

3. a 1-dimensional cell (an *edge*) is either a maximal connected portion of (the relative interior of) an edge of some triangle Δ_i, which does not intersect any other triangle, or a maximal connected portion of (the relative interior of) the intersection segment $\Delta_i \cap \Delta_j$ of two triangles, which does not meet a third triangle, and

4. a 0-dimensional cell (a *vertex*) is a vertex of a triangle, an intersection of an edge of one triangle with another triangle, or an intersection point of three distinct triangles.

To simplify the analysis, we henceforth assume that our triangles are placed in *general position* in space, meaning that no four share a point, no three intersect in more than a point, no two in more than a segment, no two edges of different triangles intersect, no triangle vertex lies in another triangle, and no triangle edge intersects another triangle in more than a point. These assumptions guarantee in particular that each of the above types of cells have the correct dimensionality. Since

our goal is to analyze the combinatorial complexity of (certain portions of) $A(G)$, this assumption involves no real loss of generality, because any degenerate layout of triangles can always be viewed as a limiting case of layouts in general position, whose combinatorial complexity dominates that of the limit configuration. Roughly, this can be argued as follows: Each triangle in the arrangement can be slightly expanded to eliminate all degeneracies involving triangle boundaries. Moreover, this expansion can be performed without changing the topology of or reducing the number of faces on any of the cell boundaries. The remaining degeneracies can be eliminated by a slight perturbation of the planes containing the triangles—such a perturbation never destroys a face, though it may create new faces. Thus we obtain an arrangement of triangles in general position in which no cell is bounded by fewer faces than it was in the original arrangement. Theorem 5.A.4 of Appendix 5.A.2, to be discussed below, essentially states that the complexity of any set of cells is dominated by the number of faces on their boundaries, implying our claim.

We will use the following additional terminology. A vertex of a triangle will be called a *corner*; an edge of a triangle will be called an *exposed edge*, and the 1-dimensional cells of $A(G)$ into which it is decomposed will be called *exposed segments*. Finally, the unqualified term *cell* will be used to denote a 3-dimensional cell of $A(G)$.

At various points in our analysis we will have to consider *planar arrangements of segments* which are, similar to the 3-dimensional case, partitions of a plane into *faces*, *edges*, and *vertices* induced by a collection of segments in that plane. As a bounded face of such an arrangement need not be simply connected, its boundary will in general consist of a single *outer component* that encloses the entire face and zero or more *islands* each enclosing a portion of the plane that does not belong to the face. An unbounded face has a similar structure, except its boundary does not have an outer component.

5.1.2 Motivation and Main Results

Our goal is to obtain sharp upper and lower bounds on the maximum (worst-case) *combinatorial complexity* of any single cell C of $A(G)$, that is the number of vertices, edges, and faces of $A(G)$ lying along the boundary of C. Consider first the following classification. A cell C of $A(G)$ is *interesting* if its closure $\overline{C}$ contains at least one exposed segment (it follows easily from the definition that, if $\overline{C}$ intersects an exposed segment, it fully contains it). All other cells of $A(G)$ are called *dull*. It is easily checked that a dull cell of $A(G)$ is (the interior of) a convex polyhedron bounded by (some of) the planes containing the triangles; its combinatorial complexity is thus only $O(n)$. Interesting cells, on the other hand, may have highly irregular shape; in fact, if n is sufficiently large, an interesting cell may approximate any open connected semi-algebraic set in $\mathbf{R}^3$, its boundary may have arbitrary genus, etc.[1] Let $\zeta(n)$ be the maximum combinatorial complexity of a single (interesting) cell, over all possible arrangements of n triangles in general position. Simple examples show that $\zeta(n)$ is $\Omega(n^2)$ (see Figure 5.2). A more sophisticated example demonstrates that $\zeta(n)$ is $\Omega(n^2\alpha(n))$ [PS87], where $\alpha(n)$ is the extremely slowly growing inverse Ackermann function. In fact, the combinatorial complexity of the *upper envelope* of the triangles in G (which is a portion of the boundary of the unbounded cell of $A(G)$, certainly an interesting cell), is always $O(n^2\alpha(n))$ [PS87]. Following the results of [PS87], as well as the work of [PSS88] and [EGS88b], mentioned below, on the two-dimensional analog of our problem, it is natural to conjecture that

$$\zeta(n) = O(n^2\alpha(n)). \qquad \textit{(Conjecture)}$$

We have been able to prove this conjecture only for a few special cases, and in general it is still an open problem. The only previously known non-trivial upper bound on $\zeta(n)$, given in [PS87], is $O(n^{3-\frac{1}{49}})$; it is proved using a fairly intricate analysis based on extremal hyper-graph theory. (It is easy, by the way, to show that the worst-case

[1] Note that, although in the simple case of triangles (or arbitrary *flat* objects) the notions of dull and interesting coincide with those of convex and non-convex, respectively, this is not necessarily the case for more general surface patches.

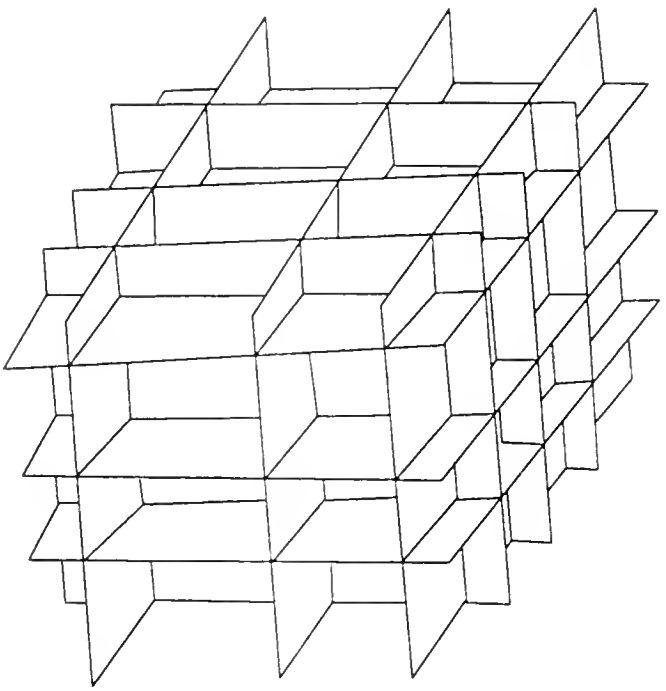


Figure 5.2: A grid-like arrangement.

combinatorial complexity of the *full* arrangement $A(G)$ is $\Theta(n^3)$.) Our results (stated below) improve and generalize this bound, and we believe that our techniques can be extended to yield a bound close to that conjectured above.

The main motivation (also discussed in [PS87,EGS87]) for studying this problem comes from motion planning. Specifically, let B be an arbitrary polyhedral object bounded by K faces, edges, and vertices and free to *translate* in three-dimensional space while avoiding stationary polyhedral obstacles bounded by a total of n faces, edges, and vertices. Using the standard technique of expanding obstacles in configuration space, it is easily seen ([PS87,EGS87]) that the (3-dimensional) space FP of all free placements of B can be represented as a union of certain (3-dimensional) cells of $A(G)$, for the collection G of $O(Kn)$ triangles obtained as the Minkowski (vector) differences between faces, edges, and vertices of the obstacles and vertices, edges, and faces of B, respectively. Moreover, given an initial placement z_0 of B, it suffices to calculate only the connected component of FP containing z_0, because no other portion of FP can be reached from z_0 by a collision-free motion. Thus, analysis of the combinatorial complexity and efficient calculation of a single cell of $A(G)$ are major components in the design and analysis of efficient algorithms for this translational motion planning problem.

Before presenting our results, let us first review the analogous (and relatively simpler) situation in two dimensions, namely the case of an arrangement A of n line segments in the plane. It is known that the worst-case combinatorial complexity of a single face of A is $\Theta(n\alpha(n))$, which is asymptotically the same as the worst-case complexity of the upper envelope of n segments [PSS88,HS86,WS88]. Also, the worst-case combinatorial complexity of all interesting (i.e., non-convex) faces of A is $O(n^{4/3}\alpha(n)^{2/3}\log^{1/3} n)$ [AEGS] (see also [EGS88b]); this bound is almost tight, as the complexity in question can be $\Omega(n^{4/3})$ [EW86]. The above upper bound is a special case of the result of [AEGS], who argue that the complexity of any m distinct faces in a planar arrangement of n segments is

$$O(m^{2/3}n^{2/3}\alpha(n)^{2/3}\log^{1/3}\frac{n^2\alpha(n)^2}{m} + n\log n).$$

It was further shown in [AEGS] that the bound can in fact be generalized to

$$O(m^{2/3}p^{1/3}\alpha(n)^{2/3}\log^{1/3}n + n\log n),$$

where p is the number of pairs of intersecting segments. (Note that it yields

$$O(p\alpha(n)^{2/3}\log^{1/3}n + n\log n)$$

when $m = \Theta(p)$, which nearly coincides with the actual $\Theta(p+n)$ complexity of the entire arrangement.) Furthermore, if the m faces are those intersected by another (fixed) segment, their total complexity is only $O(n\alpha(n))$, as has been recently shown in [EGP*88]. These results for two-dimensional arrangements are closely related to bounds that we obtain and are extensively used in our analysis. Moreover, they clearly point toward the conjecture made above.

As already mentioned, we have not been able to obtain a general near-quadratic upper bound for $\zeta(n)$. Instead, we consider here the problem of analyzing a larger quantity—the maximum total combinatorial complexity $\phi(n)$ of *all* interesting cells in any arrangement of n triangles in 3-space. Our main result is:

> **Theorem 5.2.5(c):** $\phi(n) = O(n^{7/3}\alpha(n)^{2/3}\log^{4/3}n)$; this is almost tight since $\phi(n) = \Omega(n^{7/3})$.

We thus obtain a significant improvement over the bound of [PS87], both because our bound is sharper and because it applies to the total complexity of all interesting cells, rather than that of a single one. In fact, we obtain a somewhat more general bound that depends on the number p of pairs and the number t of triples of intersecting triangles in G. Specifically, in Theorem 5.2.5(a) we show that, if $t \geq \frac{p\log^2 p}{\alpha(p)^2}$ and $p \geq n$, the total complexity of the interesting cells in an arrangement of n triangles is

$$O(p^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{4/3}p),$$

(which reduces to the previous bound by putting $p = O(n^2)$ and $t = O(n^3)$). Moreover, our analysis can be extended to obtain similar bounds on $\phi(n)$ for arrangements

of arbitrary convex flat objects (called *convex plates* in [PS87]), with the possible addition of a linear correction term to accommodate the complexity of the "exposed" boundary of these plates.

The results outlined above generalize the $O(n^{7/3}\log n)$ upper bound of [EGS88a] on the complexity of $O(n^2)$ cells in an arrangement of n *planes*. In fact, our Theorem 5.2.3 (with $t = O(n^3)$, $p = O(n^2)$) yields a generalization of the bound

$$O(m^{2/3}n\log n + n^2),$$

given in [EGS88a] for the complexity of any m cells in a plane arrangement, to arrangements of triangles. However, our result is valid only if all non-convex cells are included in the collection of desired cells, so in our case m has to be "large", i.e. at least $3n + 2p$ (see below for details and explanation of this term).

We also consider the task of actual calculation of a single cell of $A(G)$ and present a randomized algorithm, whose expected running time is $O(\zeta(n)n^{\delta})$ for any $\delta > 0$ (with the constant of proportionality depending on δ). Moreover, we demonstrate that the same algorithm is faster when applied to some special classes of arrangements. We also describe an alternative algorithm that computes all interesting cells in an arrangement of n triangles in expected time $O(n^{8/3}\alpha(n)^{2/3}\log^{41/6} n)$ and $O(n^{7/3}\alpha(n)\log n)$ space, and present a modification of this algorithm that computes any subset of cells in the arrangement. For details, refer to Section 5.5.2.

There are two special cases where both complexity bounds and algorithmic performance can be significantly improved. One case involves triangles lying in planes having only a small number f of orientations (e.g., when each triangle is parallel to one of the three coordinate planes). In this case we show that the maximum complexity of a single cell as well as that of all interesting cells is $\Theta(n^2)$ for a fixed f. More precisely, the latter complexity is $\Theta(fn^2)$ for $f = O(n^{1/3})$ and the former is $\Omega(n^2\alpha(f))$. We also provide an efficient $O((M + n^2)f\log n)$-time *deterministic* algorithm for the calculation of any subset of cells in such an arrangement, given a point in each cell, where M is the total complexity of the cell(s) being computed.

The second restricted class of arrangements contains only two types of objects: (1) arbitrary horizontal convex plates (parallel to the xy-plane) and (2) vertical rectangles (whose planes are parallel to the z axis and whose top and bottom sides are parallel to the xy-plane). In this case we show that the maximum complexity of a single cell is $\Theta(n^2\alpha(n))$.

In addition, we argue that our unmodified general algorithm computes any single cell of an arrangement of n triangles, either with only a constant number of orientations or in the second special case above, in expected time $O(n^{2+\delta})$ (for any $\delta > 0$).

The technique used to show the $O(n^{7/3}\alpha(n)^{2/3}\log^{1/3} n)$ upper bound on $\phi(n)$ is relatively simple. The proof is based on the results concerning the combinatorial complexity of many faces in an arrangement of segments in the plane, summarized above. In addition, we adapt and extend some of the technical tools developed in [EGS88b,EGS88a] for studying related problems. More specifically, supposing that bounds on the complexity of all interesting cells in two subarrangements have been obtained, we seek to establish a sharp relationship between these complexities and the complexity of all interesting cells in the arrangement formed by the union (i.e., overlay) of these two subarrangements. Such relationships (called "combination lemmas") have been derived and exploited in [EGS88b,AEGS] for the case of lines or segments in the plane and in [EGS88a] for the case of planes in 3-space. We obtain a similar relationship for arrangements of triangles (see Lemma 2.1 below), using a proof technique that is somewhat different from (and, we believe, simpler than) those employed in [EGS88b,EGS88a].

To analyze the time performance of our main algorithm we prove an important technical result ("the Slicing Theorem") which may be of independent interest. It states roughly that any collection of cells in an arrangement of triangles can be subdivided into tetrahedra without substantially increasing the total complexity of this collection.

This Chapter is organized as follows: In Section 5.2 we obtain our main results on the combinatorial complexity of all interesting cells. Section 5.3 derives the Slicing

Theorem. Section 5.4 discusses the special cases mentioned above and demonstrates the improved bounds. Algorithms for computing single and multiple cells in arrangements of triangles are presented in Section 5.5, and a discussion of some applications and open problems is given in Section 5.6. Appendix A contains some basic facts concerning the topology of cells in a three-dimensional arrangement of triangles, which are needed for our analysis. Appendix B exemplifies the usefulness of our techniques by presenting a simple alternative proof of (a variant of) the 2-dimensional combination lemma for polygonal regions used in [EGS88b].

5.2 The Complexity of All Interesting Cells

Let $G = \{\Delta_1, \ldots, \Delta_n\}$ be a collection of triangles in 3-space as in the introduction. We determine the total complexity of the interesting cells of $A(G)$ by analyzing a more general problem:

> Consider collections G of n triangles in $\mathbf{R}^3$ with at most p intersecting pairs and at most t intersecting triples, and any set P of $m \geq 2p + 3n$ points, *containing at least one point on each exposed segment* of $A(G)$; we wish to determine the maximum complexity $C(m, n, p, t)$ of all cells of $A(G)$ that contain at least one of the given points in their interior or on their boundary, where the complexity of a cell containing multiple points is to be counted *only once*, and where the maximum is taken over all such collections G and sets of points P. (Our points will be chosen so that each point lies in the closure of a unique cell.)

Note first that the number of exposed segments in $A(G)$ is $2p + 3n \leq 2\binom{n}{2} + 3n = O(n^2)$. Indeed, each endpoint of an exposed segment is either a triangle corner or an endpoint of the intersection segment of a pair of triangles. If P contains only points lying on exposed segments, one point on each segment, the *marked* cells of $A(G)$ (i.e., the cells whose closures contain those points) are precisely all the interesting cells. Hence, $C(2p+3n, n, p, t)$ serves as an upper bound on the complexity of all interesting

cells of $A(G)$, and $C(2\binom{n}{2}+3n, n, \binom{n}{2}, \binom{n}{3})$ is an upper bound on $\phi(n)$ (and, therefore, also on $\zeta(n)$). In the more general problem formulation, we consider all interesting cells plus possibly some additional dull cells.

We employ the following divide-and-conquer approach to analyzing $C(m,n,p,t)$. Partition G into two subsets G_1 and G_2 with $|G_1| = \lfloor n/2 \rfloor$, $|G_2| = \lceil n/2 \rceil$. Refer to the elements of G_1 as *red triangles*, and to those of G_2 as *blue*. Obtain recursively the "red" marked cells $R_1, \ldots, R_r$ of $A(G_1)$, and the "blue" marked cells $B_1, \ldots, B_b$ of $A(G_2)$ (so that the closure of each R_i and each B_j contains at least one point of P). For each point $p_i \in P$, let R_{r_i}, B_{b_i} be the red and the blue cells marked by p_i, respectively. Let E_i be the connected component of $R_{r_i} \cap B_{b_i}$ containing p_i, i.e., the cell of $A(G)$ marked by p_i; we refer to the cells E_i as *purple cells*. Let the total complexity of all red cells be ρ and that of all blue cells be β. Our goal is to show that the total complexity of the purple cells is at most

$$\rho + \beta + O(m^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{1/3} p + p\log p).$$

We will refer to this property, to be proved below, as the *Combination Lemma for an Arrangement of Triangles*. It generalizes a similar combination lemma obtained in [EGS88a] for planes in $\mathbf{R}^3$.

5.2.1 The Combination Lemma

Before proceeding with the analysis, let us note that the complexity of a cell C is simply the sum of the number of vertices, edges, and faces of the boundary ∂C of C. However, in order to simplify the representation of cell boundaries for our algorithms, we will count some of these features with multiplicities. Namely, imagine replacing each triangle Δ by a "puffy triangle" Δ^* which is a thin body bounded by two copies of Δ, slightly curved outward from Δ and seamed together at the (relative) boundary of Δ. If the thickness of these puffy triangles is kept sufficiently small, then our general position assumption on the triangles in G guarantees that the resulting arrangement of the puffy triangles (more precisely, the common exterior of these thin bodies)

maintains the combinatorial and topological structure of $A(G)$, except that each face of $A(G)$ now appears as two distinct faces, one on each side of the corresponding puffy triangle and, similarly, each (intersection) edge of $A(G)$ appears as four distinct edges, each (triple-intersection) vertex of $A(G)$ appears as eight distinct vertices, etc. In the subsequent analysis, we will follow this "puffy model" and take the resulting multiplicities of boundary features into account.

In Appendix A we also show that the number of edges and vertices on the boundary ∂C of a cell C is bounded by a linear function of the number of faces of ∂C plus a certain correction term which accounts for the fact that the vertices, edges, and faces along ∂C do not necessarily constitute a triangulation of that boundary and that C and/or some of its faces need not be simply connected. It is demonstrated as well that the total sum of these correction terms, over all cells in the arrangement, is only $O(n + p)$. Thus the complexity of all interesting cells can be estimated by simply bounding the total number of faces on the boundaries of such cells, and adding an $O(n + p)$ "overhead" term, which will be absorbed anyway in the bound that we will obtain. In the remainder of our discussion we will use the term "complexity of a cell" to refer exclusively to the number of its faces.

Let us now return to the partitioning of G into red and blue subcollections G_1, G_2, as above. Consider a resulting purple cell E; it is a connected component of the intersection of a red cell R and a blue cell B. Each face of E is a portion of a (red) face of R or of a (blue) face of B. We will separately bound the number of blue faces and the number of red faces of E (and sum these bounds over all purple cells). Consider red faces first. Notice that the same face of R can yield a number of purple faces that may appear in E as well as in other purple cells. However, creation of *one* purple face out of each red face is already accounted for by the overall "red" complexity ρ, so let us concentrate on the number of "extra" faces cut out of red faces—these appear either because a portion of the boundary of some blue cell B may split a red face f into a number of disconnected pieces, or because there may be two points of P in (the closure of) R belonging to different purple cells and each of these cells may have

a portion of f on its boundary. The former situation occurs only if at least one of B, f is non-convex.

We start with the red cells (i.e., those marked by P) in $A(G_1)$, and proceed to construct the final purple cells incrementally. Each step of this procedure involves adding a blue triangle $\Delta = \Delta_i$ to the subarrangement A_{i-1} of all the red and the first $i-1$ blue triangles, to obtain the next subarrangement A_i (thus $A_0 = A(G_1)$ and $A_{\lceil n/2 \rceil} = A(G)$). A "currently purple" cell $E^{(i-1)}$ (i.e., a cell of A_{i-1} marked by P) may be modified by the addition of Δ in one of several ways—it may be trimmed but remain a single cell with the same topological structure, it may remain a single cell but be cut by Δ in a way which changes the topology of its boundary, it may be split into two or more subcells, each marked by a point of P, or its boundary may acquire a new connected component (if Δ is entirely contained in $E^{(i-1)}$). (We need not be concerned with the last case though, as in such a situation the introduction of Δ has no influence on the number of red faces in purple cells.) We wish to estimate the number of *additional* red faces created on the boundary of currently purple cells (in the arrangement A_i) by the introduction of Δ. Summing the resulting bounds over all Δ_i, and repeating the argument for blue faces (starting with the blue arrangement and adding to it red triangles one at a time), we will obtain an upper bound for the purple complexity in terms of the red and blue complexities.

Consider the intersection of Δ with the current purple arrangement A_{i-1}. It appears as an arrangement $Q = Q(\Delta)$ of segments in the plane of Δ (all clipped to within Δ), each face of which corresponds to a face of some cell (actually to a pair of faces—one on each side of Δ) in the updated arrangement A_i. For convenience, the relative boundary $\partial\Delta$ of Δ is also added to Q. Let p_i be the number of segments (other than the three edges of Δ) participating in Q (in other words, the number of triangles Δ_j intersecting Δ for $j < i$). Let t_i be the number of intersecting pairs of segments of Q (again, we do not include intersections involving $\partial\Delta$). As discussed in the introduction, a bounded face of Q need not be simply connected; its boundary consists of one exterior connected component and of zero or more interior components—so

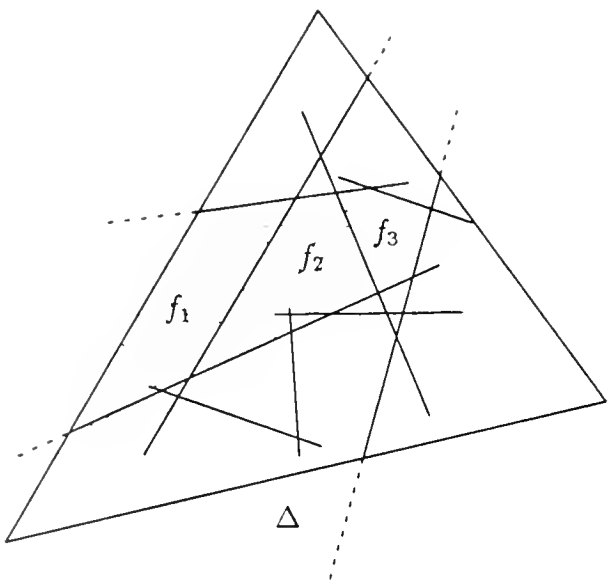


Figure 5.3: The arrangement $Q(\Delta)$ with three "purple" faces.

called *islands*.

We will refer to a face of Q incident to $\partial\Delta$ as a "boundary face." If we erase the three edges of Δ, the boundary faces fuse into the single unbounded cell of the arrangement formed solely by the intersections of triangles Δ_j (with $j < i$) with Δ, so that their total complexity, by [PSS88], is at most $O(p_i\alpha(p_i))$. (See Figure 5.3.)

Observe that a "boundary face" f (such as face f_1 in Figure 2.1) cannot split a currently purple cell E of A_{i-1} into two subcells, as f does not completely cut through E. Nevertheless, it *can* produce additional red faces. In fact, every segment e on the boundary of f that does not touch $\partial\Delta$ represents a cut of some purple face f' of A_{i-1} (note that f' need not be a red face, but only an earlier added blue face; also, f' need not be cut in two by e, but just have e reduce the number of its islands). In other words, the total number of additional red faces produced by all boundary faces on Δ is no larger than the total complexity of all those boundary faces, i.e. $O(p_i\alpha(p_i))$.

Let us now consider an internal face f of $Q(\Delta)$ (such as faces f_2, f_3 in Figure 5.3).

Recall that we restrict our attention to faces f which are contained in a currently purple cell. Since f is internal, it must either cut such a cell E into two subcells E_1, E_2 or change the topology of the boundary of E without splitting E. In the former case, if one of the two resulting subcells (say E_1) does *not* contain points of P, no additional red faces are produced because, by our choice of P, E_1 must be dull, i.e. convex (for otherwise its closure would have to contain an exposed segment, and thus also a point of P). In particular, f itself is convex and thus cuts each face of E at most once. Hence each face of E yields at most one face of E_2, as asserted. If, on the other hand, both E_1 and E_2 contain points of P, the number of new red faces created is at most proportional to the complexity of f (by an argument analogous to the case of a boundary face). Let the number of such internal "splitting" faces of $Q(\Delta_i)$ be s_i. Finally, there is the case when introduction of an internal face f does not cut E into two subcells, but only changes the topology of its boundary. Again the number of additional red faces being created is at most the number of edges of f. Let the number of such "cutting-but-not-splitting" faces of $Q(\Delta_i)$ be c_i. Hence the total number of additional red faces created by Δ_i is $O(p_i\alpha(p_i))$ plus the complexity of $s_i + c_i$ faces in a planar arrangement of p_i segments of which t_i pairs actually intersect. The latter complexity, by [AEGS], is

$$O((s_i + c_i)^{2/3} t_i^{1/3} \alpha(p_i)^{2/3} \log^{1/3} p_i + p_i \log p_i). \tag{5.1}$$

Before proceeding with the analysis, let us observe that $\sum_{i=1}^{\lceil n/2 \rceil} p_i \leq p$ since in the incremental construction the segment $\Delta_i \cap \Delta_j$, if non-empty, appears at most once, namely in $Q(\Delta_{\max\{i,j\}})$. Similarly, $\sum_{i=1}^{\lceil n/2 \rceil} t_i \leq t$. Summing (5.1) and the contributions of the boundary faces over all blue triangles Δ_i we obtain

$$O\left(\sum_{i=1}^{\lceil n/2 \rceil} \left[(s_i + c_i)^{2/3} t_i^{1/3} \alpha(p_i)^{2/3} \log^{1/3} p_i + p_i \log p_i\right]\right) =$$

$$O\left(\sum_{i=1}^{\lceil n/2 \rceil} (s_i + c_i)^{2/3} t_i^{1/3} \alpha(p)^{2/3} \log^{1/3} p\right) + O(p \log p).$$

which, by Hölder's inequality, is

$$O\left(\left[\sum_{i=1}^{\lceil n/2\rceil}(s_i+c_i)\right]^{2/3}\times\left[\sum_{i=1}^{\lceil n/2\rceil}t_i\right]^{1/3}\alpha(p)^{2/3}\log^{1/3}p\right)+O(p\log p)=$$

$$O\left(\left[\sum_{i=1}^{\lceil n/2\rceil}s_i+\sum_{i=1}^{\lceil n/2\rceil}c_i\right]^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{1/3}p\right)+O(p\log p).$$

Since P contains m points, it is impossible to make more than $m-1$ cuts, each of which splits a cell containing more than one point of P into two subcells each containing at least one point of P. Hence $\sum_i s_i \le m-1$. As to $\sum_i c_i$, we prove in Appendix A the following

> **Proposition 5.A.2:** In an incremental construction of an arrangement of triangles with p intersecting pairs, as above, the total number of faces that cut cells without splitting them is at most $O(p)$.

Recall that by assumption $m \ge 2p+3n$, so $\sum_i s_i + \sum_i c_i \le m + O(p) = O(m)$. Putting everything together, and repeating the argument for the blue faces, we see that the total increase in the number of faces of the purple cells is at most $O(m^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{1/3}p + p\log p)$. We have thus shown:

Lemma 5.2.1 (Combination Lemma for Arrangements of Triangles) *Let G be a set of n triangles in $\mathbf{R}^3$, let p and t be the number of pairs and of triples of triangles of G which intersect, and let P be a set of m points, with at least one point of P lying on each exposed segment of the arrangement $A(G)$. Partition G into two sets G_1, G_2, and denote by ρ (respectively, β) the complexity (i.e., the number of faces) of all cells of $A(G_1)$ (respectively, $A(G_2)$) marked by P. Then the total complexity of the cells of $A(G)$ marked by P is at most*

$$\rho+\beta+O(m^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{1/3}p+p\log p).$$

The recurrence relation for $C(m, n, p, t)$ that we want to develop next also depends on the following graph-theoretic lemma. For a hypergraph H, let $e(H)$ be the number of edges in H.

Lemma 5.2.2 *Given an arbitrary 3-uniform hypergraph H on n vertices, let $\{A, B\}$ be a partition of its vertices with $|A| = \lfloor n/2 \rfloor$ and $|B| = \lceil n/2 \rceil$. Let H_1 (resp. H_2) be the sub-hypergraph of H spanned by A (resp. B). Then for some choice of A and B*

$$e(H_1) + e(H_2) < \frac{1}{4}e(H).$$

Proof: For simplicity, assume for the remainder of the proof that n is even. Let A be a randomly selected set of $n/2$ vertices of H (such that all sets of size $n/2$ are chosen with equal probability). Let B be the set of remaining vertices. For an edge e in H,

$$\Pr[e \text{ in } H_1] = \Pr[e \text{ in } H_2] = \frac{\binom{n-3}{n/2-3}}{\binom{n}{n/2}} < \frac{1}{8}.$$

In particular, the expected number of edges in H_1 and H_2 together is

$$\exp[e(H_1) + e(H_2)] = e(H) \times (\Pr[e \text{ in } H_2] + \Pr[e \text{ in } H_2]) < \frac{1}{4}e(H),$$

implying the existence of a choice of A and B with $e(H_1)+e(H_2) < \frac{1}{4}e(H)$, as asserted. □

Using the above result in conjunction with the Combination Lemma, we deduce:

Theorem 5.2.3 *If G is a set of n triangles in $\mathbf{R}^3$ with p intersecting pairs and t intersecting triples and P is a set of m points, with at least one point of P lying on each exposed segment of the arrangement $A = A(G)$, the total complexity of all cells of A marked by P is*

$$C(m, n, p, t) = O(m^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{1/3} p \log n + n + p \log p \log n).$$

Proof: Observe that the total complexity of an arbitrary arrangement of triangles is proportional to the number of its vertices, which is easily seen to be $O(n+p+t)$. Thus $C(m,n,p,t)$ must satisfy the following recurrence relation:

$$C(m,n,p,t) \leq \begin{cases} a(n+p+t), & \text{if } m \geq n+p+t, \\ C(m,n/2,p_1,t_1) + C(m,n/2,p_2,t_2) + & \\ \quad b(m^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{1/3}p + p\log p), & \text{otherwise,} \end{cases}$$

where a and b are constants, and p_1, t_1, p_2, t_2 are the numbers of pairs and triples of intersecting triangles within each of the two subcollections G_1, G_2, respectively. Lemma 2.2 applied to the triple-intersection hypergraph of the triangles (i.e., the hypergraph whose nodes are triangles and which contains an edge $\{\Delta_i, \Delta_j, \Delta_k\}$ whenever $\Delta_i \cap \Delta_j \cap \Delta_k \neq \emptyset$ for distinct i, j, k) implies that there is a partitioning of G into two subsets G_1 and G_2 each of size roughly $n/2$ such that $t_1 + t_2 \leq \frac{1}{4}t$. Also trivially $p_1 + p_2 \leq p$.

We claim that

$$C(m,n,p,t) \leq dm^{2/3}t^{1/3}\log n\alpha(p)^{2/3}log^{1/3}p + en + fp\log p\log n, \qquad (5.2)$$

for some constants d, e, f, depending on a and b. First, note that if $m \geq n+p+t$, (5.2) is trivially satisfied provided $d, e, f > a$. Let us turn our attention to the general case. For the recurrence to be satisfied, the following inequality must hold:

$$dm^{2/3}t^{1/3}\log n\alpha(p)^{2/3}\log^{1/3}p + en + fp\log p\log n \geq$$
$$dm^{2/3}t_1^{1/3}\log(n/2)\alpha(p_1)^{2/3}\log^{1/3}p_1 + en/2 + fp_1\log p_1\log(n/2) +$$
$$dm^{2/3}t_2^{1/3}\log(n/2)\alpha(p_2)^{2/3}\log^{1/3}p_2 + en/2 + fp_2\log p_2\log(n/2) +$$
$$b(m^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{1/3}p + p\log p).$$

The terms linear in n cancel and $p_1, p_2 \leq p$, so it is sufficient to ensure that the following two inequalities hold:

$$dt^{1/3}\log n \geq dt_1^{1/3}\log(n/2) + dt_2^{1/3}\log(n/2) + bt^{1/3} \qquad (5.3)$$

and

$$fp\alpha(p)\log n \geq fp_1\alpha(p_1)\log(n/2) + fp_2\alpha(p_2)\log(n/2) + bp\alpha(p). \qquad (5.4)$$

First of all,

$$\begin{aligned} fp_1 \log p_1 \log(n/2) + fp_2 \log p_2 \log(n/2) + bp \log p \\ \leq\ & fp \log p \log(n/2) + bp \log p \\ \leq\ & fp \log p \log n + (b-f)p \log p. \end{aligned}$$

Thus (5.4) holds, provided that $f > b$. Turning to (5.3), an application of Hölder's inequality yields

$$t_1^{1/3} + t_2^{1/3} \leq 2^{2/3}(t_1 + t_2)^{1/3} \leq 2^{2/3}\left(\frac{t}{4}\right)^{1/3} = t^{1/3}.$$

In particular,

$$\begin{aligned} dt_1^{1/3}\log(n/2) + dt_2^{1/3}\log(n/2) + bt^{1/3} &\leq dt^{1/3}\log(n/2) + bt^{1/3} \\ &\leq dt^{1/3}\log n + (b-d)t^{1/3}, \end{aligned}$$

ensuring that (5.3) holds for any $d > b$, and thus completing the proof. □

For developing a lower bound, it will be convenient to have the following graph-theoretical result:

Lemma 5.2.4 *Let H be an undirected graph with E edges and C 3-cycles. Then $C = O(E^{3/2})$.*

Proof: Immediate from the following theorem that gives the maximum number C_r of complete subgraphs on r vertices contained in a graph with E edges (in our case, $r = 3$).

Theorem (Erdős [Erd62]):

$$\text{Let } E = \binom{s}{2} + q,\ 0 < q \leq s, \text{ then } C_r \leq \binom{s}{r} + \binom{q}{r-1}.$$

□

Having proved Theorem 5.2.3, we obtain our main results concerning all interesting cells by substituting $m = 3n + 2p$ and then observing that $p = O(n^2)$ and $t = O(n^3)$:

Theorem 5.2.5

(a) The complexity of all interesting cells in an arrangement of n triangles with p intersecting pairs and t intersecting triples is

$$O((n+p)^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{1/3}p\log n + n + p\log p\log n),$$

which reduces to $O(p^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{4/3}p)$, if $t \geq \frac{p\log^2 p}{\alpha(p)^2}$ and $p \geq n$. Moreover, for any $t \geq p \geq n$ such that $t = O(p^{3/2})$ there exists an arrangement of n triangles with $O(p)$ intersecting pairs and $O(t)$ intersecting triples so that the complexity of all interesting cells is

$$\Omega(p^{2/3}t^{1/3} + p\alpha(p) + n).$$

(b) The said complexity is also

$$O((n+p)^{2/3}p^{1/2}\alpha(p)^{2/3}\log^{1/3}p\log n + p\log p\log n + n),$$

regardless of the value of t; this is bounded by $O(p^{7/6}\alpha(p)^{2/3}\log^{4/3}p)$, provided $p \geq n$.

(c) In the worst case, $\phi(n) = O(n^{7/3}\alpha(n)^{2/3}\log^{4/3}n)$. This is almost tight, since $\phi(n) = \Omega(n^{7/3})$.

Proof: The upper bounds in (a) and (c) are immediate from the preceding analysis, while the bound in (b) easily follows from that in (a) by substituting $t = O(p^{3/2})$, which is a consequence of Lemma 5.2.4 applied to the intersection graph of the triangles (note that a triple intersection of triangles is necessarily a 3-cycle in this graph, while the converse is not true).

Turning to the lower bound in (a), let us assume $t \geq p \geq n$ and put $a = \lfloor t/p \rfloor$ and $b = \lfloor p^2/t \rfloor$. Notice that $p^2 \geq t$ by assumption, so both a and b are positive integers. To obtain the lower bound in (a), consider a set of a vertical rectangles whose bottom edges lie in the plane $z = 0$, whose top edges lie in the plane $z = b+1$,

and such that their xy projections yield a planar arrangement of a segments for which the complexity of all interesting (i.e., non-convex) faces is $\Omega(a^{4/3})$ [EW86]. Cutting these rectangles by b additional horizontal (and sufficiently large) rectangles lying in the planes $z = i$, for $i = 1, \ldots, b$, is easily seen to yield an arrangement of $n = a + b$ rectangles (that can be transformed into an arrangement of n triangles in general position) in which the total complexity of all interesting cells is

$$\Omega(a^{4/3}b) = \Omega(\frac{t^{4/3}}{p^{4/3}} \times \frac{p^2}{t}) = \Omega(p^{2/3}t^{1/3}).$$

Notice that the number of pairwise intersections in this arrangement is no more than $a^2 + ab = \frac{t^2}{p^2} + p = O(p)$ (as $t^2 = O(p^3)$ by assumption), while the number of triples of intersecting triangles is at most $a^2b \leq t$. This completes the construction of an arrangement of triangles with $O(p)$ intersecting pairs and at most t intersecting triples in which the complexity of all interesting cells is $\Omega(p^{2/3}t^{1/3})$. In addition, note that the interesting cells always have complexity $\Omega(n)$. Moreover, the construction in [PS87] of an arrangement of n triangles in which the unbounded cell has complexity $\Omega(n^2\alpha(n))$ can easily be modified to yield a similar construction with the unbounded cell having complexity $\Omega(p\alpha(p))$. Putting the three bounds together, we obtain the lower bound in (a). The lower bound in (c) follows by substituting $t = O(n^3)$ and $p = O(n^2)$. □

Remark: The reader should note the strong dependency of our bounds on the bounds for planar arrangements of segments. Any future improvements in the two-dimensional bounds (such as the recent results in [AEGS]) will most likely carry over to our analysis and yield corresponding improvements in our bounds. The exotic-looking expression $\alpha(p)^{2/3}\log^{1/3} p$ appearing in our bounds is a direct "carry-over" from the analysis of [AEGS]. Removal of this expression from the bounds in [AEGS] would also remove it from our bounds. For a more specific example, if the complexity of m faces in an arrangement of n segments intersecting in p points should be shown to be $O(m^{2/3}p^{1/3} + n\alpha(n))$, then the bound in Theorem 5.2.5(a) would go down to $O((n+p)^{2/3}t^{1/3}\log n + n + p\alpha(p)\log n)$.

5.3 The Slicing Theorem

In this Section we will prove an auxiliary result on triangulating cells in arrangements of triangles. It facilitates a general efficient algorithm for computing a single cell in such an arrangement (cf. Section 5.4). A similar technique is also used to establish better bounds and construct more efficient algorithms for some special classes of arrangements discussed below. In addition, our result provides a partial answer to the following seemingly simple question: given an arbitrary polyhedral region K in $\mathbf{R}^3$, can it be cut into a small number of (pairwise disjoint) tetrahedra? The two-dimensional analog of this problem has a satisfactory solution—the number of triangles needed to triangulate an arbitrary planar polygonal region is proportional to its complexity. In three dimensions, the problem becomes much more difficult. Some negative results are given in [OR87, Chap. 10]; our result shows roughly that the required number of tetrahedra is not much larger than the complexity of K, if K is a cell (or a collection of cells) in an arrangement of triangles. The technique that we use in the proof of the Slicing Theorem is related to that recently employed by Chazelle and Palios [CP] in obtaining similar results for general polyhedra.

Theorem 5.3.1 (Slicing Theorem) *Let K be a collection of cells in an arrangement of n triangles in 3-space, with a total of h faces. Then K can be decomposed ("triangulated") into $O(n^2\alpha(n) + h)$ tetrahedra having pairwise disjoint interiors.*

Proof: We construct the triangulation incrementally, by adding new vertical faces (to which we refer as *walls*) emanating from exposed edges of the given triangles. These vertical walls will collectively decompose the cells of K into convex subcells. Once this is done, the final triangulation is easily obtained by triangulating the boundary of each convex subcell C and connecting each resulting triangle to some fixed interior point of C. The number of final tetrahedra is clearly proportional to the total complexity of all subcells into which the walls partition K, which we now proceed to estimate.

First of all, we assume that all cells of K are interesting; the above observations imply that this involves no real loss of generality. Let $e_1, \ldots, e_{3n}$ be the exposed

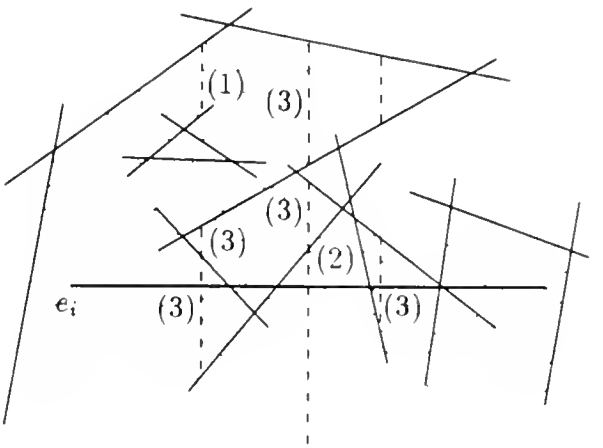


Figure 5.4: The arrangement $A(V_i)$ in the vertical plane p_i containing exposed edge e_i. Solid segments represent intersections of p_i with triangles, while three previously erected sets of walls intersecting p_i are dashed. The shaded regions represent new walls. Vertical segments are marked with their type.

edges of the triangles. For each e_i in turn we add vertical walls emanating from it, as follows. Suppose this has already been done for all e_j, $j < i$. Let p_i be the vertical plane containing e_i, and let V_i be the collection of segments in p_i, each of which is either the intersection of p_i with some triangle or a connected component of the intersection of p_i with some previously erected wall. Then the walls added at the current (ith) stage are simply all the *horizon faces* of e_i (i.e., faces incident to e_i) in the planar arrangement $A(V_i)$ which are contained in K. See Figure 5.4 for an illustration. (For convenience of exposition, we will need to assume that no exposed edge is vertical and that $p_i \neq p_j$ whenever $i \neq j$; in absence of collinear exposed edges, this can always be enforced by an appropriate rotation—collinear edges can be handled by a slight modification of this argument.)

It is easy to check inductively that the addition of all these walls results in a convex decomposition of K. Indeed, the walls added at the ith stage remove all non-convex edges occurring on e_i, and no new non-convexities are created.

We next analyze the total complexity (i.e. the number of faces) of the resulting decomposition of K. It is easily seen that this complexity is $h + O(q)$, where q is the total complexity (i.e., total number of edges) of all the vertical walls. A major obstacle in estimating q is that we have no *a priori* linear bound on the number of segments in each V_i—each triangle contributes at most one segment to V_i, but a vertical wall may intersect p_i in many segments and there are many vertical walls erected from each e_j. We overcome this difficulty as follows:

Fix an exposed edge e_i, and let V_i^0 be the collection of the (at most n) segments formed by intersecting p_i with the given triangles. By the result of [EGP*88], the total complexity of all horizon faces of e_i in $A(V_i^0)$ is $O(n\alpha(n))$. Let H be the collection of all horizon faces (within K) of e_i in $A(V_i)$ (namely the vertical walls erected from e_i). Clearly, each face in H is a subface of some horizon face in $A(V_i^0)$. Note that, for each $j < i$, all vertical walls erected from e_j intersect p_i in a collection of segments lying in the vertical line $p_i \cap p_j$ (see Figure 5.4) so that, in particular, at most one of them cuts e_i. It follows that the horizon faces in $A(V_i)$ meet e_i in at most $4n - 1$ edges, because the segments of V_i cut e_i in at most $4n - 2$ points (there are at most $3n - 1$ planes p_j with $j < i$ and at most $n - 1$ triangles not containing e_i). Since each such edge bounds at most two faces, we conclude that there are at most $8n - 2$ horizon faces in $A(V_i)$. We will subdivide each segment of $V_i - V_i^0$ into subsegments, cutting it at its points of intersection with segments of V_i^0, and work with the refined vertical segments from now on. For simplicity assume that all segments are finite; however, the argument that follows can be easily adapted to handle rays and full lines. Since each vertex of a face in H is either a vertex of a horizon face of e_i in $A(V_i^0)$ (of which there are only $O(n\alpha(n))$), or an endpoint of some vertical segment in $V_i - V_i^0$, it therefore suffices to estimate the total number of such segments which bound horizon faces of e_i in $A(V_i)$. We now add these vertical segments one at a time, thus gradually transforming the horizon faces in $A(V_i^0)$ into those in $A(V_i)$. Let V_i^j denote the current collection of segments obtained after adding j of these segments. Each segment s in $V_i - V_i^j$ bounding a final horizon face in $A(V_i)$ is of one

of the following types (note that, by construction, each endpoint of a (finite) vertical segment lies on a segment of V_i^0):

(1) s bounds the same horizon face in $A(V_i^j \cup \{s\})$ on both of its sides,

(2) s lies on the common boundary of two distinct horizon faces of $A(V_i^j \cup \{s\})$, or

(3) s bounds, on one side, a horizon face of $A(V_i^j \cup \{s\})$ and on the other side a face of $A(V_i^j \cup \{s\})$ which is no longer adjacent to e_i.

Refer to Figure 5.4. Observe that, as a new vertical segment is added to V_i^j to obtain V_i^{j+1}, the type of the remaining segments may change; in fact, it is easily verified that the only transitions possible are (1) $\rightarrow$ (2), (2) $\rightarrow$ (3), and (1) $\rightarrow$ (3). We proceed to transform the horizon faces of $A(V_i^0)$ into those of $A(V_i)$ by first adding all segments of $V_i - V_i^0$ which at the time of addition have type (1). (As noted, no transition can revert a segment to type (1).) Each such segment reduces the number of "islands" (i.e., connected components of the boundary) of some horizon face by 1. But the total number of islands in all horizon faces is at most n, because each segment of V_i^0 appears in at most one island, and (on the assumption that no vertical segment is infinite) each island must contain a segment of V_i^0. Thus the total number of type (1) segments added is at most n. A similar argument shows that the number of type (2) segments, which we add after the type (1) segments, is also $O(n)$, because addition of each such segment increases the number of horizon faces by 1, and there are only $O(n)$ such faces in the final $A(V_i)$. The remaining vertical segments all have type (3) and will never change their type again. Let V_i^* be the union of V_i^0 with all type (1) and type (2) vertical segments. Since $|V_i^*| = O(n)$, it is still the case that the complexity of all horizon faces of e_i in $A(V_i^*)$ is $O(n\alpha(n))$. Now add the vertical segments of type (3). Each such segment s, in its turn, chops off some portion f_s of a horizon face, which becomes disconnected from e_i. Moreover, since we add only segments that actually appear on the boundaries of the final horizon faces in $A(V_i)$ and vertical segments are pairwise disjoint, each portion f_s is chopped off only once

(namely, these portions have pairwise disjoint interiors) and, since f_s is a polygonal region, it must contain a vertex of a horizon face in $A(V_i^*)$ other than either of the endpoints of s. Hence the number of portions f_s, and thus also the number of type (3) segments s, is $O(n\alpha(n))$. Repeating the analysis for each e_i, we conclude that the total number of faces in the decomposition of K is $h + O(n^2\alpha(n))$, as asserted. As discussed above, a final triangulation of this decomposition completes the proof of the theorem. □

Corollary 5.3.2 *Let K be a collection of q cells in an arrangement of n triangles in $\mathbf{R}^3$, having a total of h faces. Then K can be decomposed into $q + O(n^2)$ convex polyhedra with pairwise disjoint interiors, such that their total complexity is only $h + O(n^2\alpha(n))$.*

Proof: It is sufficient to demonstrate that in the above construction the number of resulting subcells does not exceed q by more than $O(n^2)$, since the second claim follows immediately from the proof of the Slicing Theorem. Recall that a vertical wall w erected at a given step of the construction is a face in the arrangement induced in its plane by intersections with all triangles and all vertical walls erected in previous steps. Hence w cuts the subcell C in which it is contained into at most two new subcells (note that it is possible for w to change the topology of the boundary of C without cutting C into two subcells). Hence introduction of w increases the number of subcells by at most 1. However, there are only $O(n)$ vertical walls erected from each exposed edge, thus the total number of convex polyhedra obtained by the construction does not exceed q by more than $O(n^2)$. □

Corollary 5.3.3 *The collection K of all interesting cells in an arrangement of n triangles in 3-space can be decomposed into $O(n^2)$ convex polyhedra with pairwise disjoint interiors, such that their total complexity is only $h + O(n^2\alpha(n))$, where h is the complexity of K.*

Proof: Immediate from Corollary 5.3.2, since there are at most as many interesting cells as there are exposed segments, i.e. $O(n^2)$. □

Remarks: (1) While the Slicing Theorem is intuitively plausible, the proof is not at all trivial. An open problem is to generalize the theorem to the case of an arbitrary 3-dimensional semi-algebraic set K, namely to show that K can be decomposed into a collection of simple, connected cells (e.g., such as in Collins' cylindrical algebraic decomposition [Col75]), whose cardinality is roughly the same as the combinatorial complexity of ∂K. A partial affirmative result of this sort is indicated in [CEG*88] for the collection K of *all* 3-cells in an arrangement of spheres; the analysis given there appears to be extendible to arbitrary algebraic surfaces, provided, however, that we still consider all cells of the arrangement. A related open problem is to obtain generalizations of the Slicing Theorem to arrangements of simplices in higher dimensions.

(2) Observe that the key point of the proof of the Slicing Theorem is the assertion that, given a planar arrangement A of n "old" segments and an unspecified number of "new" segments, the total complexity of the faces of A cut by a segment e is still $O(n\alpha(n))$, provided the new segments (i) cross e at most $O(n)$ times, (ii) do not intersect among themselves, and (iii) start and end at points of old segments or "at infinity". It is easy to verify that the proof given above applies to this slightly more general situation as well.

(3) Note that even when the number p of pairs of intersecting triangles in the arrangement is $o(n^2)$, the number of cells in the resulting decomposition of K can still be quadratic (this is the case, for example, in the arrangement of n long and thin horizontal rectangles, $n/2$ of which lie in the plane $z = 0$, while the remaining $n/2$ lie in the plane $z = 1$, as shown in Figure 3.2; here in fact $p = 0$).

5.4 Special Cases

In this Section we will discuss two restricted classes of arrangements of triangles for which the worst-case complexity of a single cell (or of all interesting cells) is easier to bound.

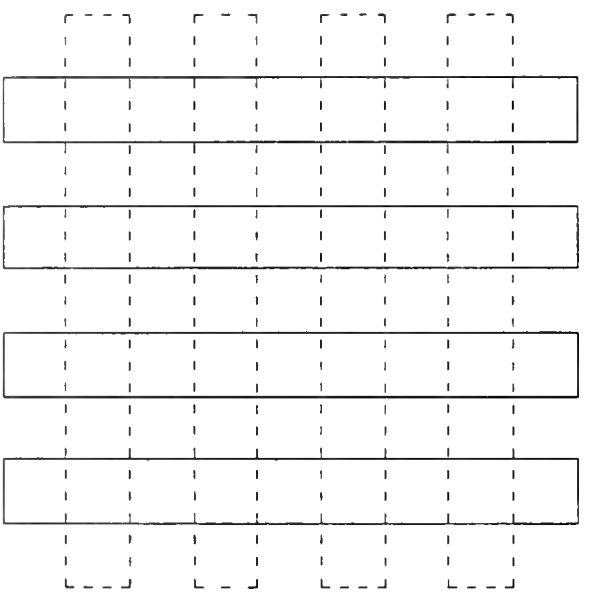


Figure 5.5: A collection of disjoint rectangles in which our "slicing" produces $O(n^2)$ polyhedra. Solid rectangles lie in the plane $z = 1$, dashed ones in $z = 0$.

5.4.1 Triangles with Few Orientations

We first consider arrangements of n triangles in which the planes containing the triangles have at most f distinct orientations. (Note that in the following analysis f need not be constant.) A (very restricted) example of such an arrangement would be when each triangle lies in a plane parallel to one of the $f = 3$ coordinate planes. In what follows, we will assume however, without loss of generality, that no triangle lies in a plane parallel to the z axis. Corollary 5.3.3 yields a decomposition of all interesting cells in such an arrangement into $O(n^2)$ polyhedra. Being convex, each of these polyhedra is bounded by at most two faces contained in planes having any specific orientation, so it is bounded by at most $2f$ faces lying in the original triangles. The remaining faces lie in the vertical walls added in the construction of the Slicing Theorem; thus they do not contribute to the complexity of the (undecomposed) interesting cells (except by cutting other faces into subfaces). Hence, the total number of non-vertical faces in all polyhedra lying in interesting cells is $O(fn^2)$, which is an upper bound on the total number of faces in all interesting cells. This proves

Theorem 5.4.1 *The complexity of all interesting cells in an arrangement of n triangles lying in planes with f distinct orientations is at most $O(fn^2)$.*

As for a corresponding lower bound, it is easy to obtain a variant of the construction of [EW86] for a lower bound on the complexity of many faces in arrangements of lines which involves f^3 segments in the plane, having only f distinct orientations, such that the complexity of all interesting (i.e., non-convex) faces of their arrangement is $\Omega(f^4)$. Assume $f \leq (n/2)^{1/3}$. Placing $\lfloor n/2f^3 \rfloor$ disjoint translates of such an arrangement side by side in the plane produces an arrangement of at most $n/2$ segments having f distinct orientations, in which the complexity of all interesting faces is $n/2f^3 \times \Omega(f^4) = \Omega(fn)$. Now consider a 3-dimensional arrangement of $n/2$ vertical rectangles whose lower edges lie in the plane $z = 0$, whose upper edges lie in the plane $z = n/2 + 1$, and whose vertical projections induce the planar arrangement of $n/2$ segments described above, together with $n/2$ large horizontal rectangles lying in the

planes $z = i$, for $i = 1, \ldots, n/2$. It is easy to see that the total complexity of the interesting cells in this arrangement of n rectangles is at least $n \times \Omega(fn) = \Omega(fn^2)$, but only $f + 1$ distinct plane orientations were used. Moreover, the rectangles can be easily replaced by triangles without reducing the combinatorial complexity of the interesting cells. Thus we have shown

Theorem 5.4.2 *The worst-case complexity of all interesting cells in an arrangement of n triangles lying in planes with f distinct orientations is at least $\Omega(fn^2)$ for $f = O(n^{1/3})$.*

A similar argument, based on the construction of a planar arrangement of n segments in which the unbounded face has complexity $\Omega(n\alpha(n))$ [WS88], gives

Theorem 5.4.3 *The worst-case complexity of a* single *cell in an arrangement of n triangles lying in planes with f distinct orientations is at least $\Omega(n^2\alpha(f))$.*

Corollary 5.4.4 *In an arrangement of n triangles lying in planes with a constant number of distinct orientations, the worst-case complexity of any single cell (or of all interesting cells) is $\Theta(n^2)$.*

5.4.2 The Horizontal-Vertical Case

The second special case concerns arrangements where only two types of convex plates are allowed: arbitrary (convex) horizontal polygons (i.e., parallel to the xy-plane) and vertical rectangles (i.e., rectangles with two sides parallel to the z axis). Let n be the total number of plates (both horizontal and vertical) in the arrangement and let q be the total complexity of all horizontal plates. The lower bound construction of Theorem 5.2.5 applies in this case, showing that the total complexity of all interesting cells in this restricted class of arrangements is still $\Omega(n^{7/3})$. We now use a variant of the decomposition given by the Slicing Theorem to prove a tight $\Theta(n^2\alpha(n) + q)$ bound on the worst-case complexity of a single cell in such an arrangement. Recall

that the only type of non-convexity that occurs in an interesting cell is along an exposed segment. We distinguish two types of interesting cells:

(1) those whose boundary contains only vertical exposed segments (i.e., those parallel to the z axis) and

(2) those whose boundary contains horizontal exposed segments (and possibly some vertical ones).

We will argue that the complexity of a single cell of type (1) is $O(n\alpha(n))$, while the *total* complexity of type (2) cells is $O(n^2\alpha(n)+q)$, concluding that the complexity of any one cell in such an arrangement is $O(n^2\alpha(n)+q)$.

Consider a cell C of type (1). Clearly, it cannot be unbounded. Moreover, the absence of horizontal exposed segments on ∂C implies that C is in fact a right vertical prism whose height is a maximal vertical open segment extending from a "floor plate" Δ_f to a "ceiling plate" Δ_c and whose bases are identical faces in the planar arrangements $Q(\Delta_f)$ and $Q(\Delta_c)$ induced in Δ_f and Δ_c, respectively, by their intersections with the vertical rectangles. $Q(\Delta_f)$ and $Q(\Delta_c)$ each contain at most n segments, so the complexity of a single face in either of the two arrangements is $O(n\alpha(n))$. Since C is a prism, its complexity is proportional to the complexity of either of its bases, implying our first claim.

There are two types of horizontal exposed segments on the boundaries of type (2) cells: portions of the horizontal edges of vertical rectangles and portions of edges of the horizontal plates. We will restrict our attention to the latter; the former are handled similarly. Let p be the plane containing some horizontal plate Δ. Intersections of vertical rectangles with p induce a planar arrangement $Q = Q(\Delta)$ in p of at most n segments. Refine the type (2) cells by adding a *wall* filling each face of Q that lies outside of Δ and is incident to the relative boundary $\partial\Delta$ of Δ. Repeating the same procedure for every horizontal plate (and every horizontal edge of a vertical rectangle, within the horizontal plane containing it), we obtain a decomposition of all cells of type (2) into subcells. Note that all horizontal exposed segments have been

eliminated, so each subcell is of type (1) (and thus is a right vertical prism). We will now argue that the bases of subcells contained in type (2) cells have total complexity $O(n^2\alpha(n))$, so that the total complexity of these subcells is bounded by $O(n^2\alpha(n))$ as well; this would complete the proof of the upper bound as the only elements of type (2) cells not accounted for in the combined complexity of their subcells is that of horizontal exposed segments and their incident faces and vertices, and there are $O(n^2+q)$ exposed segments altogether, with each exposed segment incident to exactly two faces and two vertices.

Consider a subcell C' of a type (2) cell. Let f and c be the floor and ceiling bases of C' contained in horizontal planes p_f and p_c, respectively. By construction, $\partial C' - f \cup c$ must consist of portions of the vertical rectangles extending all the way from f to c. In particular, any cross section of C' by a horizontal plane p is a face in the planar arrangement $Q(p)$ of segments induced in p by its intersections with the vertical rectangles. Therefore f (respectively, c) is the union of one or more such faces in an analogous arrangement $Q(p_f)$ in p_f (respectively, $Q(p_c)$ in p_c), which is identical to $Q(p)$ except for the addition of the edges of the plate contained in p_f (resp. c). Observe that, by construction, the closure of either f or c (or both) must contain a horizontal exposed segment, for otherwise C' would coincide with an original cell of type (1). Let us suppose that it is f that meets some horizontal exposed edge. Thus f coincides with the union of some horizon faces of the boundary $\partial\Delta_f$ of the convex plate Δ_f in the arrangement $Q(p_f)$. Moreover, it is easily checked that each such horizon face in $Q(p_f)$ lies in the base of at most two subcells (in the floor of one cell and in the ceiling of another). However, the results of [EGP*88] are easily seen to imply that the horizon of any closed convex curve in a planar arrangement of n segments has complexity $O(n\alpha(n))$; thus, the combined complexity of the bases of all subcells of type (2) cells is $O(n^2\alpha(n))$, as asserted.

The lower bound of $\Omega(n^2\alpha(n) + q)$ for the complexity of a single cell is easily obtained by a slight modification of the construction of Theorem 5.2.5. Thus we have shown

Theorem 5.4.5 *In an arrangement of n horizontal convex plates and vertical rectangles, the worst-case complexity of a single cell is $\Theta(n^2\alpha(n)+q)$, where q is the total number of corners of the horizontal plates.*

5.5 Algorithms

Here we come to the issue of actually building castles in the air, as promised in the title of this Chapter. Namely, we are interested in computing one or more cells in an arrangement A of n triangles in $\mathbf{R}^3$. Note that it is not difficult to compute the *full arrangement* A in time $\Theta(n^3)$ by modifying the algorithm of [EOS86] for constructing the arrangement of n *planes*. Thus our goal is to attain subcubic performance in computing a single cell (or all interesting cells). The previous results of [PS87] are non-algorithmic, and our combinatorial analysis cannot be immediately translated into an efficient algorithm either. In Section 5.5.1 we describe a randomized algorithm for computing a single cell in an arbitrary arrangement of n triangles; its expected time complexity is asymptotically nearly optimal. Moreover, the same algorithm takes only $O(n^{2+\delta})$ time, for any $\delta > 0$ (with the constant of proportionality depending on δ), when applied to arrangements of vertical rectangles and horizontal convex plates discussed in the previous section (assuming that each horizontal plate has constant complexity), or to arrangements of n triangles lying in planes with a constant number of distinct orientations. We then comment on the difficulty of extending our algorithm to compute many cells, and outline an alternative, less efficient but still subcubic algorithm for obtaining all interesting cells, which is based on the recent results of [EGH*88]. Finally, we present a customized deterministic algorithm for arrangements of triangles with few distinct plane orientations (as in Section 5.4).

All of our algorithms use a common data structure to represent cells in three-dimensional arrangements of triangles. A cell is represented by the list of its boundary components, starting with the explicitly marked outer boundary (unless the cell is unbounded) and followed by zero or more "islands", all linked to one another (the

exact nature of this linkage will be discussed below). Each boundary component is stored in the form of an "adjacency structure", where each face is given by its outer boundary (unless the face is unbounded) plus zero or more islands, properly linked to one another, each of which is in turn represented as a circular list of incident edges and vertices. Conceptually, we store each face twice (corresponding to the two sides of the triangle), leading to similar multiplicities in the representation of edges and vertices, as in the puffy model discussed in Section 5.2.1. This ensures that every boundary element is incident to a *unique* cell (duplicate elements are still linked among themselves; in fact an actual implementation need not store them as physically distinct entities). Thus each (copy of an) edge is incident to exactly two faces, a triple-intersection vertex is incident to three faces and three edges, etc. The precise implementation of such a data structure can be done along the lines suggested in [TDS] or in [DL87]; however we will not be concerned here with such implementation details.

5.5.1 Calculating a Single Cell

Our algorithm proceeds as follows:

Input: A set G of n triangles in $\mathbf{R}^3$, a point p, and an "efficiency parameter" $\delta > 0$.

Output: The cell of $A(G)$ containing p, represented by the incidence structure discussed above.

(1) Select a random sample R of r triangles (where r is a constant, depending on δ, to be determined later).

(2) Construct the full arrangement $A(R)$ (using any convenient, even brute-force, method).

(3) By exhaustive search, identify the cell C_R of $A(R)$ that contains p.

(4) Subdivide C_R into open tetrahedra as in the Slicing Theorem (again, using brute force).

(5) Let t_o be the tetrahedron containing p. For each triangle $\Delta \in G$ compute the convex polygon $\Delta \cap t_o$, and let G_{t_o} be the collection of such (triangulated) non-empty polygons. Apply the algorithm recursively to (G_{t_o}, p, δ) to obtain a cell C_o. If C_o is a bounded cell of $A(G_{t_o})$, stop—C_o is the desired cell of $A(G)$ containing p. Otherwise apply the trimming step (8) below to $C_{t_o} = C_o$ (denoting the resulting trimmed subcell containing p by $\widehat{C}_o$).

Repeat steps (6)–(8) for each of the remaining tetrahedra $t \neq t_o$ in the decomposition of C_R and proceed to step (9).

(6) As for t_o, compute for each triangle $\Delta \in G$ the convex polygon $\Delta \cap t$, and let G_t be the collection of all such non-empty (triangulated) polygons.

(7) Apply the algorithm recursively to (G_t, p_∞, δ) where p_∞ is "the point at infinity", to obtain the unbounded cell C_t of $A(G_t)$.

(8) Trim C_t along the faces of t and discard the outer portion of the cell; this may split C_t into several subcells all lying within t and incident to the boundary ∂t of t.

(9) Reconstruct the cell C of $A(G)$ containing p by starting with $\widehat{C}_o$ and "growing" it as follows: Let a *window* of a (trimmed) cell C' of $A(G_t)$ be a face of C' on ∂t introduced by the trimming step (8). Clearly each window is shared by two cells lying in adjacent tetrahedra. A window is *transparent* if it is not contained in a triangle of G. Starting with $\widehat{C}_o$, locate all cells sharing transparent windows with it and "glue" them to $\widehat{C}_o$ along the windows. Repeat until the resulting cell C has no unpaired transparent windows. This is the desired cell.

Why does the algorithm work? Observe that the cell C_G of $A(G)$ containing p is completely contained in the cell C_R of $A(R)$ containing p, so that the algorithm correctly restricts its computation to C_R. Notice that if p does not lie in the unbounded cell C_{t_o} of $A(G_{t_o})$ then C_G must coincide with C_o, in which case no further computation is necessary and the algorithm correctly returns C_o. Assume $p \in C_{t_o} = C_o$ which

is unbounded. If a point x lies in $C_G - \widehat{C}_o$, there is a path σ in C_G connecting it to p. A traversal of σ from p to x starts off in $\widehat{C}_o$ and proceeds to visit trimmed cells of $A(G_t)$ (for various t), moving from cell to cell through transparent windows. Moreover, each cell C crossed by σ must be a portion (within the corresponding tetrahedron t) of the unbounded cell C_t of $A(G_t)$, because C can be reached from outside t. Hence x will be in the cell built by the algorithm. So the constructed cell is no smaller than C_G and, trivially, it cannot be larger than C_G either. Thus the cell the algorithm computes is precisely C_G.

We next analyze the expected running time of the algorithm. Since r is a constant, all steps of the algorithm besides recursive invocations and the trimming and reconstruction steps can be performed in overall linear time. Denoting by $|C|$ the complexity of a cell C, observe that trimming the unbounded cell C_t of $A(G_t)$, for some tetrahedron t, can be performed in time proportional to $|C_t|$, as it is sufficient to check each feature of C_t against each face of t. The time required for the reconstruction step is proportional to the total complexity of all transparent windows, because this step can be accomplished by simultaneously tracing the common face of each pair of adjacent tetrahedra using the incidence structure of the trimmed cells. Determining whether C_o is the unbounded cell of $A(G_{t_o})$ can be accomplished in constant time, since the representation we use explicitly stores the outer component (if any) of a cell boundary. Hence the total time complexity of all steps of the algorithm, excluding recursive calls, is bounded by $An + B\sum_t |C_t|$, for some positive constants A, B depending on r, with the summation taken over all tetrahedra t contained in C_R. Observe that a small perturbation of $\bigcup_t G_t$ (that moves tetrahedra away from each other) produces an arrangement of $\sum_t |G_t|$ triangles in which the boundary of the unbounded cell is precisely $\bigcup_t \partial C_t$. In particular, $\sum_t |C_t| \leq \zeta(\sum_t |G_t|)$. It immediately follows that the time complexity of the algorithm exclusive of recursive calls is bounded above by $O(n+\zeta(\sum_t |G_t|))$. Now $\sum_t |G_t|$ is easily seen to be bounded by Dn, for some constant D depending on r. Therefore, the algorithm requires $O(\zeta(Dn))$ overhead and recurs on K problems $(G_{t_i}, p_\infty, \delta)$, where K is the total number of tetra-

hedra contained in C_R (by the Slicing Theorem, $K \leq J\zeta(r)$ where J is a constant *independent* of r), and t_i varies over all such tetrahedra (including t_o). By the ϵ-net theory of [HW87] (see also [Cla87]), with high probability, each tetrahedron meets at most $\frac{an\log r}{r}$ triangles, for some constant $a > 0$ independent of r. Moreover, with no extra overhead we can verify that this does indeed hold for our sample R; if not we simply discard R and try another sample, until we hit one which has the ϵ-net property. It easily follows that the expected number of such iterations is a constant (depending on r). Hence, in expected linear time, the algorithm produces a sample for which $|G_{t_i}| \leq \frac{an\log r}{r}$, for all i. In particular, denoting by $T(n)$ the expected running time of the algorithm, we obtain the following recurrence for $T(n)$:

$$T(n) \leq \begin{cases} KT(\frac{an\log r}{r}) + O(\zeta(Dn)), & \text{if } n > r, \\ O(r^3), & \text{otherwise.} \end{cases}$$

Define $\gamma = \limsup_{h\to\infty} \frac{\log\zeta(h)}{\log h}$. Recall that $\zeta(n)$ has been shown to be $\Omega(n^2\alpha(n))$ and $O(n^{7/3}\alpha(n)^{2/3}\log^{4/3} n)$, so $2 \leq \gamma \leq 7/3$. Hence, for any $\delta > 0$, there is a choice of a constant b such that the term $O(\zeta(Dn))$ is bounded above by $bn^{\gamma+\delta/2}$ for all n (with b depending on δ and r). Thus the recurrence reduces to

$$T(n) \leq \begin{cases} KT(\frac{an\log r}{r}) + bn^{\gamma+\delta/2}, & \text{if } n > r, \\ O(r^3), & \text{otherwise.} \end{cases}$$

which has a solution $T(n) \leq Hn^{\epsilon}$ for any $\epsilon > \max\{\gamma + \frac{\delta}{2}, \log K/\log\frac{r}{a\log r}\}$ (with H depending on r, δ and ϵ). Notice that the second quantity can be made not to exceed $\gamma + \frac{\delta}{2}$ by choosing a sufficiently large r, as $K \leq J\zeta(r)$. Choosing such an r for the given input value of δ, we obtain $T(n) = O(n^{\gamma+\delta})$, with the constant depending on δ. In particular, we have shown:

Theorem 5.5.1 *Given a fixed $\delta > 0$, a single cell in an arrangement of n triangles in $\mathbf{R}^3$ can be computed in randomized expected time $C_\delta n^{\gamma+\delta}$, with the constant C_δ depending on δ, where $\gamma = \limsup_{k\to\infty} \frac{\log\zeta(k)}{\log k} \in [2, 7/3]$.*

Corollary 5.5.2 *A single cell in an arrangement of n triangles can be computed in randomized expected time $O(n^{7/3+\delta})$, for any $\delta > 0$, with the constant of proportionality depending on δ.*

Remark: Note that the expectation in the time bound of our algorithm is over the random selection of the sample R, and *not* over any distribution of the input. Our algorithms thus have the same asymptotic expected complexity for any general arrangement of n triangles. Notice that, should the conjecture posed in the introduction be demonstrated to hold, the above theorem would immediately provide an $O(n^{2+\delta})$ expected time algorithm for computing any single cell of the arrangement.

We also note that the polygons passed to recursive invocations of the algorithm will not in general be triangles. However, it is easy to see that, independently of the level of recursion on which such a polygon is created, it can be represented as the intersection of an original triangle with a single tetrahedron. Hence such a polygon can have no more than 7 sides, and the presence of non-triangles can be compensated for by using $\zeta(5n)$ instead of $\zeta(n)$ in our argument (as any convex polygon with up to 7 sides can be cut into at most 5 triangles), and by replacing the constant a by $5a$.

Remark: Several recent techniques (due to Clarkson [Cla87], Chazelle and Friedman [CF88], and Matoušek [Mat88]) provide, in certain special cases, tools either for a deterministic construction of a triangulation of space with properties similar to those discussed above, or for obtaining such a decomposition in which each cell is cut by only $O(\frac{n}{r})$ objects, rather than $O(\frac{n}{r}\log r)$. We do not know whether these techniques can be adapted to the problem at hand, so as to make the above algorithm deterministic or further improve its time complexity.

Finally, observe that the time complexity analysis of the above algorithm relies on the following facts:

1. The complexity of any cell in arrangement of n triangles is bounded by a function $\zeta(n)$ such that a cell in the arrangement formed by any subset of r triangles can be decomposed into $O(\zeta(r))$ tetrahedra.

2. With high probability, the number of triangles cutting any tetrahedron that is missed by the r sample triangles is at most $\frac{an}{r}\log r$, for some constant a independent of r and n.

3. The intersection of a tetrahedron with a triangle is a convex polygon of constant complexity.

Thus, in a situation where the above conditions hold (perhaps with a different function ψ taking the place of ζ), the algorithm will correctly compute the desired cells in expected time $O(\psi(n)n^{\delta})$, for any $\delta > 0$. In particular, we have:

Theorem 5.5.3 *Given a fixed $\delta > 0$, a single cell in an arrangement of n triangles lying in planes with a constant number f of distinct orientations can be computed in randomized expected time $C_{\delta,f}n^{2+\delta}$, with the constant $C_{\delta,f}$ depending on δ and f.*

Theorem 5.5.4 *Given a fixed $\delta > 0$, a single cell in an arrangement of n horizontal triangles and vertical rectangles can be computed in randomized expected time $C_{\delta}n^{2+\delta}$, with the constant C_{δ} depending on δ.*

Observe that Theorem 5.5.3 is superceded by Theorem 5.5.6 of Section 5.5.3 which, for a constant f, provides a faster and more general *deterministic* algorithm for computing portions of such arrangements.

5.5.2 Calculating Many Cells

An attempt to extend the algorithm of Section 5.5.1 to calculate an arbitrary collection of cells of an arrangement of triangles (e.g. all interesting cells) faces the following technical problem: our algorithm takes advantage of the fact that the desired single cell is contained in a single cell of the arrangement $A(R)$ of the random sample of triangles. This implies that the number of tetrahedra that require further processing is only $O(r^{7/3}\alpha(r)^{2/3}\log^{4/3} r)$ (in fact, $O(\zeta(r))$), which leads to a recurrence with a favorable time complexity. In contrast, when calculating many cells, there is no *a*

priori sharp bound on the overall complexity of the cells of $A(R)$ which contain the desired cells of $A(G)$—in the worst case all cells of $A(R)$ could be involved—so that no comparable recurrence relation can be obtained. Hence, although our algorithm can be easily adapted to calculate a collection of cells, we do not have sharp worst-case bounds on its expected running time in this case.

Instead, we use the following alternative approach. Edelsbrunner et al. [EGH*88] recently described a randomized procedure which, given a planar arrangement Q of n segments, preprocesses it in expected time $O(n^{5/3}\log^{5/2} n)$ and space $O(n^{4/3}\alpha(n)\log n)$ so that, given any query point x, one can calculate the face of Q containing x in time $O(n^{1/3}\log^{11/2} n + k)$, where k is the complexity of that face. Using this procedure, a subset of the cells of $A(G)$ can be computed as follows.

Consider first the task of calculating all interesting cells. We begin by obtaining a point on each exposed segment (in overall $O(n^2)$ time). Consider each of these points y in turn, and let Δ be a triangle containing y. The algorithm proceeds to trace ∂C, where C is the cell containing y, as follows. It first obtains the face f containing y in the arrangement induced in Δ by intersections with the remaining triangles, using the procedure of [EGH*88], and then moves to adjacent faces of ∂C, which share an edge with f and lie in other triangles, obtains each of them, and repeats this procedure until the entire boundary component is traced this way. One then needs to "hop" from one boundary component of C to another. This is done by "linking" triangles in the following fashion: In the preprocessing stage, for each of the $3n$ triangle corners, one computes (by brute force) the triangle that lies directly below it (if any) and the point of intersection of this triangle with a vertical line through the corner. Given such a point z lying directly below a corner c, one proceeds (again, by brute force) to the boundary of the face containing z (in the induced planar arrangement) and then follows the boundary until the first vertex c' of that face is encountered; then c' is linked to c and vice versa. Corners that have no triangle below them are linked to the "vertex at infinity". The desired "boundary-hopping" is then accomplished by examining all vertices in the computed boundary component and

following the links from any of them to vertices that have not yet been encountered, which then serve as starting points y in new boundary components. Consider an interior boundary component K of a bounded cell. Since the lowest vertex of K is a triangle corner, it is easy to verify that *some* link connects K to either the outer boundary component or to an interior component whose lowest point is below that of K. This immediately implies that all interior components are (possibly indirectly) linked to the outer component. This, in particular, allows one to pass from one boundary component of a cell to another until all have been traced. The argument in the case of the unbounded cell is similar, with the "vertex at infinity" playing the role of the outer component of the boundary. It is easy to verify that linear time is sufficient to compute each of the links by brute force, thus only quadratic additional time suffices to guarantee that no component of the desired boundary will be left unvisited.

Consider next the general case where we want to calculate an arbitrary collection of cells, each specified by a point x in its interior. For each such cell C, we first identify some point y on its boundary ∂C. This can be done in linear time per cell by choosing y to be the first point of intersection of the downward-directed vertical ray emanating from x with any triangle; if there is no such point, let y be the "vertex at infinity"—C in this case is the unbounded cell and the algorithm will then start at the vertex at infinity and follow all precomputed links from it to boundary components of C (as in the discussion just presented).

If the number m of points marking cells is very large, this linear overhead per marker x may be too expensive. In this case we can locate the desired points y as follows. Extend each triangle into a full plane, and apply the (randomized) algorithm of [EGS88a] which calculates the plane lying immediately below each of these m points in expected time $O(m^{3/4-\delta}n^{3/4+3\delta}\log^2 n)$ for any $\delta > 0$. We thus obtain m points y, each lying below a corresponding given point x on one of the n planes containing the triangles. Using the precomputed data structures of [EGH*88] for fast face queries, we compute, for each y, a point on the boundary of the face f containing y in the

arrangement induced in the plane of the corresponding triangle, thereby obtaining a point on the boundary of the cell containing y (and, therefore, x). This requires $O(n^{1/3}\log^{11/2} n)$ expected time per query.

Having obtained in this manner an "anchor vertex" on the boundary of each of the desired cells, we can then continue as in the case of all interesting cells described above. We omit further details concerning this extension. Thus we obtain

Theorem 5.5.5 *Any collection of cells of $A(G)$, defined by specifying a set of m points marking the desired cells, can be computed in expected time*

$$O(m^{3/4-\delta}n^{3/4+3\delta}\log^2 n + (M+m)n^{1/3}\log^{11/2} n + n^{8/3}\log^{5/2} n)$$

and $O(m^{3/4-\delta}n^{3/4+3\delta} + M + n^{7/3}\alpha(n)\log n)$ working storage, for any $\delta > 0$ (with the constant of proportionality depending on δ), where M is the total complexity of the desired cells. If the number m of marking points is small, a more straight forward approach will compute the marked cells in expected time

$$O(mn + Mn^{1/3}\log^{11/2} n + n^{8/3}\log^{5/2} n)$$

using $O(M + n^{7/3}\alpha(n)\log n)$ space. A somewhat simpler algorithm computes all interesting cells of $A(G)$ in expected time

$$O(n^{8/3}\alpha(n)^{2/3}\log^{41/6} n)$$

and space $O(n^{7/3}\alpha(n)\log n)$.

Remark: Notice that the amount of work required for computing an arbitrary subset of cells is largely dominated by the preprocessing time, which in turn requires nearly as much time as that for computing all interesting cells. Therefore, it may be assumed that all interesting cells are to be calculated anyway and one is facing the problem of computing some additional dull (i.e., *convex*) cells of an arrangement of triangles, where each cell is specified by a point in it.

5.5.3 A Deterministic Algorithm for Arrangements with Few Orientations

In this Section we sketch an efficient $O((m + M + n^2)f \log n)$-time deterministic algorithm for computing a subset of m cells in an arrangement of n convex plates lying in planes with only f distinct orientations, where M is the total complexity of the cells being computed. For simplicity of presentation we will assume that the plates are in fact triangles and that no more than three triangles meet at a common point, but a more general situation can be handled just as easily. Each desired cell C is identified by a point $x \in C$. Given x, the algorithm locates a vertex v_o on ∂C, using a technique similar to that in Section 5.5.2, but admitting a much simpler implementation. Specifically, the plane defined by a triangle and lying immediately below x can be determined in $O(f \log n)$ time by performing f binary searches, one in each collection of parallel planes. Once the point y lying in such a plane immediately below x is located, a point on ∂C is easily obtained by one $O(f \log n)$-time ray-shooting query (described below) in that plane, and a vertex on ∂C can be obtained after two more queries of this kind). Then it traces out all edges and vertices of ∂C by starting at v_o and repeatedly "shooting" along an edge of ∂C incident to the "current" vertex v and thus discovering a new vertex w of ∂C. This gives us two new edges incident to w and we shoot along them to discover further vertices.

Two specific problems arise in this approach. First, how can each "shot" be performed in $O(f \log n)$ time? Shooting along exposed segments is easily implemented by precomputing the answers to all possible queries, while each ray-shooting query along a line of intersection of two plates reduces to a two-dimensional ray-shooting in some arrangement induced in either plate by its intersections with the remaining plates. By assumption, each such arrangement A_i in plate Δ_i is formed by at most $f - 1$ overlaid subarrangements, each of which consists of a collection of parallel segments. Moreover, there are only $f - 1$ possible direction for the shooting ray. Thus, for each subarrangement and each shooting direction we can prepare a data structure that supports $O(\log n)$ shooting queries using, for example, the technique

of [ST86]. It is easily checked that this approach guarantees $O(f \log n)$ query time while the preprocessing can be accomplished for all n plates in $O(fn^2 \log n)$ time and $O(fn^2)$ storage. (This also takes care of the initial step of moving from each given point x to a vertex on the boundary of the cell containing it.)

The second problem that arises is handling clusters of edges lying in ∂C but not connected to each other. Such situations can occur if there are "islands" in faces of ∂C, or if ∂C is not connected. We have dealt with the latter case in Section 5.5.2 by linking each triangle corner to (a vertex lying in) the triangle directly below it to facilitate "hopping" from one component of ∂C to another. The former situation is approached similarly except that the linking is performed in each planar arrangement A_i induced in the plate Δ_i by intersections with the remaining plates. For each segment endpoint c in A_i, we locate the segment s lying immediately below c (in Δ_i). Then c is linked to the vertex of A_i lying on s nearest to the point of s immediately below c. If the "linking" steps are carried out using ray-shooting, the total overhead can be shown not to exceed $O(fn^2 \log n)$.

Theorem 5.5.6 *Given a set of m marking points, the cells marked by these points in an arrangement of n triangles lying in planes with f distinct orientations can be computed in deterministic time $O((m+M+n^2)f \log n)$ and $O(M+fn^2)$ storage, where M is the total complexity of the marked cells. In particular, if no two markers lie in the same cell, the time complexity of the algorithm reduces to $O((M+n^2)f \log n)$.*

5.6 Discussion and Open Problems

The preceding analysis leaves one major open problem, namely the settling of our conjecture that $\zeta(n) = \Theta(n^2\alpha(n))$. In addition, we note the following extensions and applications of our results:

(1) As already mentioned, the analysis of Section 5.2 easily extends to yield a similar upper bound for the complexity of all non-convex cells in an arrangement of n arbitrary convex (flat) plates in 3-space (with an additive correction term that accounts

for the complexity of the plate boundaries). We omit the straightforward details of this extension.

(2) We believe that our results can be generalized to higher dimensions; specifically we conjecture that the maximum number of facets (highest-dimensional faces) of all non-convex cells in an arrangement of n d-simplices in $(d+1)$-dimensional space is close to $O(n^{d+1/3})$. (Note that the lower bound construction can be generalized to any number of dimensions, yielding a lower bound of $\Omega(n^{d+1/3})$ for this complexity.) The technique used in [EGS88a] to obtain such an extension for the case of hyperplanes may be useful for proving this conjecture.

(3) The technique we used to prove the Combination Lemma can be adapted to yield simpler proofs for other combination lemmas, such as those given in [EGS88b, EGS88a]. Appendix B exemplifies this claim by providing a simple proof for a variant of the two-dimensional combination lemma of [EGS88b] for faces in arrangements of segments.

(4) Expanding upon our motion-planning application, we note that our analysis shows that the accessible portion C of the configuration space FP of a general, not necessarily convex, polyhedron B with K faces translating amidst polyhedral obstacles having n faces altogether, has

$$O((Kn)^{7/3}\alpha(Kn)^{2/3}\log^{4/3}(Kn))$$

combinatorial complexity, and that it can be calculated in expected time $O((Kn)^{7/3+\delta})$ (more precisely, in time $O(\zeta(Kn)(Kn)^{\delta}))$, for any $\delta > 0$. However, to actually plan a motion between two given placements z_1, z_2 of B, it may not even be necessary to calculate the entire component C of FP containing these placements. In this regard we claim that if B can translate from z_1 to z_2, then it can always do so along a polygonal path having only $O((Kn)^2)$ turns, and that in the worst case (at least for a constant K) that many turns are necessary. The claim immediately follows from the fact that C can be cut into $O((Kn)^2)$ convex polyhedra, as shown in Section 5.3 (Corollary 5.3.2 to the Slicing Theorem). An example where $\Omega(Kn^2)$ turns are necessary (for some non-convex B) is not difficult to construct. However, we do not know

how to calculate such a path efficiently, without obtaining first the entire component C. Further extensions and applications of the combinatorial results obtained above to motion planning problems are discussed in a forthcoming paper [AS].

(5) An interesting by-product of the Slicing Theorem and the preceding remark is that the techniques of [SS88] for coordinated motion planning for two independent robots can be applied to obtain an $O(n^{14/3})$ (actually, $O(\zeta(n)^2)$) algorithm for coordinating the motions of two independent translating polyhedra, each having $O(1)$ complexity, amidst stationary polyhedral obstacles with a total of n faces. See [SS88] for more details.

(6) Let us conclude by considering again the Conjecture posed in the introduction. Theorem 5.2.3 suggests a generalized version of the conjecture which states that the complexity of any m cells of $A(G)$ is

$$O(m^{2/3}t^{1/3}\alpha(p)^{2/3}\log^{1/3} p \log n + n + p \log p \log n).$$

where p (resp. t) is the number of pairs (resp. triples) of intersecting triangles in G. In particular, putting $m = 1$, $p = O(n^2)$, and $t = O(n^3)$ in this conjectured bound would yield $\zeta(n) = O(n^2 \log^2 n)$. Does this bound on $\zeta(n)$ (slightly weaker than originally conjectured) hold?

Appendices

5.A Topology of Arrangements of Triangles

In this Appendix we derive several basic properties of the topological structure of an arrangement of n triangles in $\mathbf{R}^3$. We begin with the analysis of the number of cutting-but-not-splitting faces, required in the proof of the Combination Lemma and proceed to justify our practice of bounding the complexity of a cell by the number of its faces. Our arguments make use of the "puffy" model of the arrangement, mentioned in Section 5.2, and further discussed below.

5.A.1 The Genus of Cell Boundaries

In this Section we will analyze the topological structure of the union of a family of triangles in $\mathbf{R}^3$ in order to bound the number of "cutting-but-not-splitting" faces in the proof of the Combination Lemma.

For the following analysis, it will be useful to regard the boundary of a cell in an arrangement of triangles in $\mathbf{R}^3$ as a compact orientable two-manifold. To this end, we replace each triangle Δ in the arrangement by a "puffy triangle" Δ^* which is a thin nearly flat body bounded by two surfaces, one on each side of Δ, slightly deformed away from one another. Δ^* has the same edges and vertices as Δ, and is sufficiently thin so that the combinatorial pattern of intersections of the puffy triangles is identical to that of the original arrangement (such choice of thicknesses is always possible under the general position assumption). Cells in the new arrangement are connected components of the common exterior (i.e., complement of the union) of all puffy bodies. However, it is easy to check that now cell boundaries are compact orientable two-manifolds (without boundary). Note that in this puffy model, every original face occurs in the new arrangement exactly twice, each (pairwise intersection) edge—four times, each (triple-intersection) vertex—eight times, and each vertex that is the intersection of an exposed edge with another triangle—twice. Triangle corners and exposed segments are not duplicated.

Let us recall some facts from elementary algebraic topology (see [GP74] for reference). For a cell complex X, let $\chi(X)$ be its Euler characteristic. It has the property that, for any two cell complexes X and Y

$$\chi(X \cup Y) = \chi(X) + \chi(Y) - \chi(X \cap Y) \tag{5.5}$$

(where $X \cap Y$ is a sub-complex of X and Y). We will require the following standard facts: $\chi(S^1) = 0$, $\chi(S^1 \times I) = 0$, $\chi(S^2) = 2$, $\chi(D^2) = 1$, where $I = [0, 1]$ is the unit interval, S^1—a circle, S^2—a sphere, D^2—a closed disk, and $S^1 \times I$—a cylinder (more precisely, the surface of a bounded cylinder without the "lids"). In particular, notice that Euler characteristic is additive when X and Y are disjoint or intersect along a set

of disjoint circles and and/or cylinders. Also recall that the Euler characteristic of a compact connected orientable two-manifold (without boundary) is equal to $2(1-g)$, where g is its genus.

We now return to arrangements of triangles. Consider a triangle Δ cutting an arrangement $A = A(G)$ of n other triangles. The intersection of Δ with A is a planar arrangement $Q = Q(\Delta)$ of (at most) n segments all contained within Δ. Keeping in mind the "puffy" model of the arrangement, observe that each face f of Q corresponds to two faces in the arrangement $A' = A(G \cup \{\Delta\})$, and that the introduction of this pair of faces changes some cell C of A (either by splitting it into two subcells, or by just modifying its boundary ∂C). Our goal is to measure the effect of adding f on the genus of ∂C. For the remainder of this discussion, we will assume that Δ *does* intersect some triangle of G, for otherwise it simply adds an extra component (of genus 0) to ∂C without changing the rest of ∂C. Let us denote by g (respectively, g') the genus summed over all connected components of ∂C (respectively, $\partial C'$, where C' is the one or two cells produced by cutting C with f), and by c (respectively, c') the number of such connected components. As we will only be interested in the differences $\delta c = c' - c$ and $\delta g = g' - g$, we may assume that the quantities g, c, g', and c' refer only to the components of ∂C and $\partial C'$ met by f. Also notice that, modulo the above assumption, the Euler characteristic of ∂C can be computed as $\chi(\partial C) = 2(c-g)$ (simply by summing χ over all components of ∂C). component of ∂C separately and observing that the Euler characteristic is additive here as the components are disjoint by definition). A similar identity holds for $\chi(\partial C')$. Two cases are possible: either

(i) f touches the relative boundary $\partial\Delta$ of Δ, in which case the introduction of f cannot split C into two subcells, or

(ii) f lies within the relative interior of Δ, in which case C may or may not be split by f into two disjoint cells.

Recall that the boundary of f consists of a *single* outer component and zero or more *islands*, all enclosed by the outer boundary and enclosing mutually disjoint regions

(that lie outside of f).

We will consider case (i) first. Let the outer boundary of f be denoted $\partial_{out} f$. Suppose that f has $i \geq 0$ islands and that $\partial_{out} f - \partial\Delta$ consists of $b \geq 0$ connected components. Since we have assumed that Δ intersects some triangles of G, at least one of i, b is not zero. In the puffy model, f can be identified with a sphere with $2i + b$ open discs deleted (two discs are deleted for each island—one on either side of f—and one disk for every connected component of $\partial_{out} f - \partial\Delta$). Using (5.5), we obtain

$$\chi(f) = \chi(S^2) - (2i + b)\chi(D^2) + (2i + b)\chi(S^1) = 2 - (2i + b).$$

Now observe that the boundary $\partial C'$ of the new cell C' can be obtained by removing b disks (one for each contact of $\partial_{out} f$ with ∂C) and i cylinders (one for each island of f) from ∂C and "gluing" the remaining portion to f along the $2i + b$ "seam" circles. The Euler characteristic of ∂C with b disks and i cylinders removed is easily seen to be

$$\chi(\partial C) - i\chi(S^1 \times I) - b\chi(D^2) + (2i + b)\chi(S^1) = \chi(\partial C) - b.$$

Applying (5.5) once again, we obtain the Euler characteristic of $\partial C'$:

$$\chi(\partial C') = (\chi(\partial C) - b) + \chi(f) - (2i + b)\chi(S^1) = \chi(\partial C) + 2(1 - i - b).$$

Recalling the relation between the genus and the Euler characteristic of a compact orientable two-manifold, we conclude $2(c' - g') = 2(c - g) + 2(1 - i - b)$, or $\delta g = \delta c + i + b - 1$. Since c' (the number of components of $\partial C'$ met by f) is 1 and $1 \leq c \leq i + b$, it follows that $1 - i - b \leq \delta c \leq 0$ and thus $0 \leq \delta g \leq i + b - 1$.

Let us now consider the case (ii) where f is an internal face of Δ with i islands, so that f has two sides that are not directly connected to each other. More precisely, each side of f is a topological disk with i open disks removed, so

$$\chi(f) = 2[\chi(D^2) - i\chi(D^2)] = 2(1 - i).$$

Once again, to obtain the boundary $\partial C'$ of the new cell(s) we have to delete $i + 1$ cylinders from ∂C (one for each component of the boundary of f) and glue it to f

along the $2(i+1)$ "seam" circles. The Euler characteristic of ∂C with $i+1$ cylinders removed is $\chi(\partial C)$. "Gluing" it to f along the $2(i+1)$ circles, we obtain

$$\chi(\partial C') = \chi(\partial C) + \chi(f) = \chi(\partial C) + 2(1-i).$$

Recalling the expression for the Euler characteristic of the components of ∂C and $\partial C'$ met by f, we obtain $2(c'-g') = 2(c-g) + 2(1-i)$, so that $\delta g = \delta c + i - 1$. Here again, c' is either 2 or 1 depending on whether f splits C into two cells or not. (Clearly, if C is split in two, $c' = 2$. Otherwise, consider the connected component K of $\partial C'$ containing one side of f. If it includes the opposite side of f, $c' = 1$, as asserted. If it does not, observe that K is a compact connected oriented two-manifold (without boundary); thus its removal disconnects $\mathbf{R}^3$ into an interior and an exterior components (see, for example, [Moi77, Chap. 26]). As K does not include the other side of f, two points lying sufficiently close to each other on opposite sides of f are separated by $K \subset \partial C'$, making it impossible for both of them to lie in the same cell, and thus contradicting the assumption that C was not split in two.) On the other hand, c can be as high as $i+1$ or as low as 1 (f has one outer boundary and i islands; the extreme cases correspond to all of them lying in different components of ∂C or all lying in the same component). Hence $-i \leq \delta c \leq 1$ and $-1 \leq \delta g \leq i$. Moreover, if f does not split C into two cells, $c' = 1$ and the above relation becomes $\delta g = i - c$. In particular, we have shown that each cutting-but-not-splitting internal face f must be of one of the following two types:

(1) $i = 0$ and $\delta g = -1$, or

(2) $i > 0$.

To summarize the cases: introduction of a face f (in $Q(\Delta)$) either reduces the total genus by one or increases it by (at most) the number of its islands (if f is a boundary face of $Q(\Delta)$, it may also cause an additional increase in the total genus by as much as the number of components of $\partial_{out} f - \partial\Delta$ less 1). However, the number of islands over all faces of $Q(\Delta)$ is easily seen to be bounded by the total number p_Δ

of segments in $Q(\Delta)$ (since a segment cannot appear in two islands) and the number of components of $\partial_{out} f - \partial\Delta$ over all faces f in $Q(\Delta)$ is bounded by $2p_\Delta$. Hence:

Theorem 5.A.1 *The change in the total genus of all cell boundaries in an arrangement A of triangles, caused by adding a triangle Δ to A, is between $3p_\Delta$ and $-k$, where k is the number of internal faces of $Q(\Delta)$, having no islands, whose introduction does not split a cell of A into two subcells, and p_Δ is the number of segments in $Q(\Delta)$.*

Thus, summing these changes in an incremental construction of $A(G)$, as in the proof of the Combination Lemma in Section 5.2, we obtain:

Proposition 5.A.2 *In an incremental construction of an arrangement of triangles, as carried out in the proof of the Combination Lemma, the total number of faces that cut cells without splitting them is at most $O(p)$, where p is the number of pairs of triangles that have non-empty intersection.*

Proof: We will count the number of such faces in a construction that starts with no triangles and proceeds to add first all red and then all blue triangles, one at a time. This clearly provides an overestimate on the desired quantity. The total genus of the boundaries of all cells is always non-negative and the genus of an empty set (the boundary of $\mathbf{R}^3$—the only cell in an arrangement of no triangles) is zero. It is decreased by one by the introduction of each type (1) internal face. On the other hand, the previous theorem implies that the overall increase in the total genus is proportional to the total number of segments in all arrangements $Q(\Delta)$, which is to say, $O(p)$. Hence the total number of faces for which $\delta g = -1$ is $O(p)$. As islands cannot be shared among faces of $Q(\Delta)$, there are at most p_Δ faces of type (2) in $Q(\Delta)$, for a grand total of $O(p)$. The conclusion follows. □

Theorem 5.A.3 *The genus summed over the boundaries of all cells in an arrangement of triangles in which p pairs intersect is at most $O(p)$.*

Proof: Immediate from Theorem 5.A.1. □

5.A.2 Euler's Relationship for Cell Boundaries

In this Section we will justify our practice of bounding the complexity of a cell by the number of its faces. Consider a single connected component K of a cell boundary. By using the puffy model, we ensure that K is a compact connected orientable 2-manifold, and thus we can apply standard topological techniques to relate the number of faces, edges, and vertices in K to its Euler characteristic and genus (see, for example, [GP74]). Since faces of K are not necessarily simply connected, we must account for this as well. Let v_K, e_K, f_K be the number of vertices, edges, and faces of K, respectively. If all faces of K were simply connected, the definition of the Euler characteristic $\chi(K)$ of K would yield

$$v_K - e_K + f_K = \chi(K).$$

Consider a non-simply connected face—its boundary consists of a single outer component and zero or more islands. Notice that introduction of i cuts transforms an i-island face into a simply-connected one, where a cut is a simple arc connecting a vertex of an island to a vertex of the outer boundary of the face. Thus all non-simply-connected faces of K can be eliminated by increasing the number of edges in K by the total number i_K of "islands" on the faces of K. Hence

$$v_K - (e_K + i_K) + f_K = \chi(K) \text{ or } v_K + f_K = \chi(K) + e_K + i_K. \tag{5.6}$$

Notice that all vertices of K besides triangle corners have degree three or more, while triangle corners have degree two. Denoting the number of such corners on K by c_K and summing the degree over all vertices, we obtain

$$2c_K + 3(v_K - c_K) \leq 2e_K \text{ or } 3v_K \leq 2e_K + c_K. \tag{5.7}$$

Conditions (5.6) and (5.7) together imply

$$e_K \leq c_K + 3f_K - 3\chi(K) \text{ and } v_K \leq c_K + 2f_K - 2\chi(K).$$

Recall that the Euler characteristic of a compact connected orientable manifold is $2(1 - g_K)$, where g_K is the genus of K, yielding

$$e_K < c_K + 3f_K + 6g_K,$$

and a similar bound for v_K. Summing these inequalities over all connected components K of ∂C (with C being a fixed cell or any collection of cells), we obtain

$$e_{\partial C} \leq c_{\partial C} + 3f_{\partial C} + 6g_{\partial C} = O(f_{\partial C} + n + p)$$

and

$$v_{\partial C} \leq c_{\partial C} + 2f_{\partial C} + 4g_{\partial C} = O(f_{\partial C} + n + p),$$

where $g_{\partial C}$ and $c_{\partial C}$ refer to the total genus of all components of ∂C (in the sense of the last Section) and the total number of triangle corners on these components, respectively. The last equality follows from Theorem 5.A.3 and the observation that there are $3n$ triangle corners in the whole arrangement. Thus we have shown:

Theorem 5.A.4 *The total combinatorial complexity of a collection of cells in an arrangement of n triangles in space is $O(F + p + n)$, where F is the total number of faces in the boundaries of these cells and p is the number of intersecting pairs of triangles.*

5.B Other Combination Lemmas—An Example

In this Appendix we demonstrate the strength of the technique that we used in proving the Combination Lemma (Lemma 5.2.1) by employing it to obtain a simple proof of a variant of the combination lemma of [EGS88b] for faces in arrangements of segments in $\mathbf{R}^2$.

Lemma 5.B.1 *Consider two arrangements of segments,* red *and* blue, *with a total of n segments. Assume that no two segments intersect in more than a point. Let*

$\mathcal{B} = \{B_1, ..., B_s\}$ (resp. $\mathcal{R} = \{R_1, ..., R_t\}$) be a family of faces in the red (resp. blue) arrangement. Let P be a set of k points $\{p_1, ..., p_k\}$, such that each point p_i lies in $B_{s_i} \cap R_{t_i}$ for some unique s_i and t_i. For each $i = 1, ..., k$, let E_i denote the face of the combined arrangement (i.e., the connected component of $B_{s_i} \cap R_{t_i}$) containing p_i. Then the complexity of the "purple" family $\{E_i\}$ is at most $\beta + \rho + O(k+n)$, where β, ρ are the complexities of $\mathcal{B}$, $\mathcal{R}$, respectively, and the complexity of a face containing two or more points of P is still counted only once.

Proof: Let n_r and n_b be the number of red and blue segments, respectively. We follow the general approach of the proof of Lemma 5.2.1, but the analysis here is considerably simpler. That is, starting with the red arrangement, we will incrementally superimpose the blue segments on it, thereby trimming and subdividing the red faces in $\mathcal{R}$ until they assume their final "purple" shape. In the process we will bound the number of additional *red* edges that are created on purple boundaries. Repeating the same process, starting with the blue family $\mathcal{B}$ and incrementally transforming it to the purple family, will provide an estimate on the number of additional blue edges bounding the E_i's. Together, these bounds give an upper bound on how much the complexity of $\{E_i\}$ exceeds $\rho + \beta$.

We thus begin with the red family $\mathcal{R}$ and add blue segments, one by one. At any step during this process, we refer to the polygonal regions of the current planar map containing points of P as "currently purple". Consider the next blue segment e to be added to the arrangement. We will refer to a maximal segment of the intersection of e with a currently purple face as a *blue fragment.* Clearly the next currently purple family may be obtained by adding all blue fragments of e to the currently purple regions and then considering the regions of the resulting planar map which contain points of P. Consider adding blue fragments to the current arrangement one at a time. Let b be a blue fragment (contained in some currently purple face Q) being added in the current step. By construction, the relative interior of b does not meet any currently purple face boundaries. Therefore, b can be classified into exactly one of the following five classes:

(i) b cuts Q into two subfaces, each containing points of P;

(ii) b cuts off a portion of Q containing no point of P;

(iii) b connects two components of ∂Q, without fully cutting Q;

(iv) b meets ∂Q at only one of its endpoints;

(v) b is a full blue segment lying entirely in the interior of Q.

(Note that the class of a fragment is determined *at the time of insertion.*) We now proceed to estimate the total number of blue fragments in each class over the course of construction and the amount of additional red complexity that they create. In case (v), introduction of b clearly has no effect on the red complexity of currently purple faces, so let us restrict our attention to blue fragments of types (i)-(iv). Each fragment of type (i) or (iii) creates at most two new red edges (as it meets ∂Q exactly twice). There are at most $k-1$ type (i) fragments, as each such fragment further refines the partition of P into sets of points lying in distinct currently purple cells. As each type (iii) fragment connects two "islands" of a currently purple cell and no segment (either blue or red) can lie in two islands, type (iii) fragments occur at most $n-1$ times. In particular, there are at most $n_r - 1$ type (iii) fragments, connecting two points of red segments and thus increasing the red complexity by 2, and at most n_b fragments of type (iii) linking a point on a previously inserted blue segment to one on a red segment, each increasing the red complexity by 1. The fragments of type (iii) that connect two points of previously inserted blue segments do not affect the red complexity. As for type (ii) fragments, note that the endpoints of b lie on *different* edges of ∂Q, as b is a line segment and no two segments overlap. Each such edge is cut by the endpoint of b into two subedges, one of which lies outside of the newly truncated currently purple cell, and thus no additional red edges are created. Turning to type (iv) fragments, each of them creates at most two new red edges out of the edge that it meets, thereby increasing the red complexity by no more than one. However, the endpoint of b that does not lie on ∂Q must be, by construction, an

endpoint of a blue segment, so that there are at most $2n_b$ type (iv) blue fragments. Thus, once all blue segments have been added, the overall increase in the number of red edges in all purple regions is at most

$$2(k-1) + 2(n_r - 1) + n_b + 2n_b = 2k + 2n_r + 3n_b - 4.$$

Repeating the analysis for the increase in blue complexity, we conclude that the total complexity of all purple faces is at most

$$\beta + \rho + 4k + 5n - 8,$$

as asserted. □

Bibliography

[AA87] T. Asano and T. Asano. Voronoi diagram for points in a simple polygon. In *Discrete Algorithms and Complexity: Proc. Japan-US Joint Seminar*, pages 51–64, Academic Press, 1987.

[AE84] F. Aurenhammer and H. Edelsbrunner. An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern recognition*, 17:251–257, 1984.

[AEGS] B. Aronov, H. Edelsbrunner, L. J. Guibas, and M. Sharir. Improved bounds on the complexity of many faces in arrangements of segments. Manuscript.

[AS] B. Aronov and M. Sharir. The union of convex polyhedra and translational motion planning. In preparation.

[AT86] T. Asano and G. T. Toussaint. Computing the geodesic center of a simple polygon. In D. S. Johnson, A. Nozaki, T. Nishizeki, and H. Willis, editors, *Perspectives in Computing: Discrete Algorithms and Complexity, Proceedings of Japan-US Joint Seminar*, pages 65–79, June 1986.

[Aur87] F. Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM J. Computing*, 16:78–96, 1987.

[Aur88] F. Aurenhammer. Voronoi diagrams—A survey. November 1988. Manuscript.

[BO79] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers*, C-28:643–647, 1979.

[Bro79] K. Q. Brown. Voronoi diagrams from convex hulls. *Information Processing Letters*, 9(5):223–232, 1979.

[BS88] A. Baltsan and M. Sharir. On the shortest paths between two convex polyhedra. *J. ACM*, 35:267–287, 1988.

[CD85] L. P. Chew and R. L. Drysdale, III. Voronoi diagrams based on convex distance functions. In *Proc. of the ACM Symp. on Computational Geometry*, pages 235–244, 1985.

[CEG*88] K. Clarkson, H. Edelsbrunner, L. J. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and surfaces. In *Proc. 29th IEEE Symp. of Foundations of Computer Science*, pages 568–579, 1988.

[CF88] B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. In *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 539–549, 1988.

[Cha82] B. Chazelle. A theorem on polygon cutting with applications. In *Proc. 23rd Symp. on Theory of Computing*, pages 339–349, 1982.

[Che89] L. P. Chew. Constrained Delaunay triangulations. *Algorithmica*, 4:97–108, 1989.

[Cla87] K. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.

[Col75] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *2nd GI Conference on Automata Theory and Formal Languages*, pages 134–183, Springer, 1975.

[CP] B. Chazelle and L. Palios. Triangulating a non-convex polyhedron. In preparation.

[DL87] D. P. Dobkin and M. J. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. In *Proc. 3th ACM Symp. on Computational Geometry*, pages 86–99, 1987.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer, Heidelberg, 1987.

[EGH*88] H. Edelsbrunner, L. J. Guibas, J. Hershberger, R. Seidel, M. Sharir, J. Snoeyink, and E. Welzl. Implicitly representing arrangements of lines or of segments. In *Proc. 4th ACM Symp. on Computational Geometry*, pages 56–69, 1988.

[EGP*88] H. Edelsbrunner, L. J. Guibas, J. Pach, R. Pollack, M. Sharir, and R. Seidel. Arrangements of curves in the plane: Topology, combinatorics and algorithms. In *Proc. 15th Int. Colloq. on Automata, Languages and Programming*, pages 214–229, 1988.

[EGS87] H. Edelsbrunner, L. J. Guibas, and M. Sharir. *The Upper Envelope of Piecewise Linear Functions: Algorithms and Applications.* Tech. Report 333, Comp. Sci. Dept., Courant Institute, November 1987. To appear in *Discrete Comput. Geom.*

[EGS88a] H. Edelsbrunner, L. J. Guibas, and M. Sharir. The complexity of many cells in arrangements of planes and related problems. In *Proc. 13th Int. Symp. on Mathematical Programming*, page 147, Tokyo, Japan, 1988.

[EGS88b] H. Edelsbrunner, L. J. Guibas, and M. Sharir. The complexity of many faces in arrangements of lines and of segments. In *Proc. 4th ACM Symp. on Computational Geometry*, pages 44–55, 1988.

[EOS86] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Computing*, 15:341–363, 1986.

[Erd62] P. Erdős. On the number of complete subgraphs contained in certain graphs. *Publ. Math. Inst. Hungar. Acad. Sci.*, 7:459–464, 1962.

[ES86] H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. *Discrete Comput. Geom.*, 1:25–44, 1986.

[Eve74] B. Everitt. *Cluster Analysis.* Wiley, New York, 1974.

[EW86] H. Edelsbrunner and E. Welzl. On the maximal number of edges of many faces in an arrangement. *J. Comb. Theory*, Ser. A 41:159–166, 1986.

[For85] S. J. Fortune. Fast algorithms for polygon containment. In *Proc. 12th Int. Colloq. Automata, Languages, Programming*, pages 189–198, 1985.

[For87] S. J. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987.

[GHL*87] L. J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan. Linear time algorithms for visibility and shortest path problems inside a triangulated simple polygon. *Algorithmica*, 2:209–233, 1987.

[GJPT78] M. R. Garey, D. S. Johnson, F. P. Preparata, and R. E. Tarjan. Triangulating a simple polygon. *Information Processing Letters*, 7(4):175–179, 1978.

[GP74] V. Guillemin and A. Pollack. *Differential Topology.* Prentice-Hall, 1974.

[GS78] L. J. Guibas and R. Sedgewick. A dichromatic framework for balanced trees. In *Proc. 19th IEEE Symp. on Foundations of Computer Science*, pages 8–21, 1978.

[HS86] S. Hart and M. Sharir. Nonlinearity of Davenport-Schinzel sequences and of generalized path-compression schemes. *Combinatorica*, 6:151–177, 1986.

[HW87] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

[Kir79] D. G. Kirkpatrick. Efficient computation of continuous skeletons. In *Proc. 20th IEEE Symp. on Foundations of Computer Science*, pages 18–27, 1979.

[Knu73] D. E. Knuth. *Sorting and Searching: The Art of Computer Programming III.* Addison-Wesley, Reading, Mass., 1973.

[Lee78] D. T. Lee. *Proximity and Reachability in the Plane.* PhD thesis, Coordinated Science Laboratory, University of Illinois at Urbana, 1978.

[LL86] D. T. Lee and A. K. Lin. Generalized Delaunay triangulations for planar graphs. *Discrete Comput. Geom.*, 1:201–217, 1986.

[LP84] D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14(3):393–410, 1984.

[LS87a] D. Leven and M. Sharir. Intersection and proximity problems and Voronoi diagrams. In J. T. Schwartz and C. K. Yap, editors, *Algorithmic and Geometric Aspects of Robotics*, pages 187–228, Erlbaum Associates, Hillsdale, NJ, 1987.

[LS87b] D. Leven and M. Sharir. Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams. *Discrete Comput. Geom.*, 2:9–31, 1987.

[LW79] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Comm. ACM*, 22(10):560–570, 1979.

[Mat88] J. Matoušek. Construction of ϵ-nets. 1988. Manuscript.

[Mil64] J. Milnor. On the Betti numbers of real varieties. *Proc. Amer. Math. Soc.*, 15:275–280, 1964.

[Moi77] E. E. Moise. *Geometric Topology in Dimensions 2 and 3.* Springer, 1977.

[OR87] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, 1987.

[PS85] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer, New York, 1985.

[PS87] J. Pach and M. Sharir. *The Upper Envelope of Piecewise Linear Functions and the Boundary of a Region Enclosed by Convex Plates: Combinatorial Analysis.* Tech. Report 279, Comp. Sci. Dept., Courant Institute, March 1987. To appear in *Discrete Comput. Geom.*

[PSR] R. Pollack, M. Sharir, and G. Rote. Computing the geodesic center of a simple polygon. *Discrete Comput. Geom.* To appear.

[PSS88] R. Pollack, M. Sharir, and S. Sifrony. Separating two simple polygons by a sequence of translations. *Discrete Comput. Geom.*, 3:123–136, 1988.

[SH75] M. I. Shamos and D. Hoey. Closest-point problems. In *Proc. 16th IEEE Symp. on Foundations of Computer Science*, pages 151–162, 1975.

[Sha85] M. Sharir. Intersection and closest pair problems for a set of planar discs. *SIAM J. Comput.*, 14:448–468, 1985.

[SS88] M. Sharir and S. Sifrony. Coordinated motion planning for two independent robots. In *Proc. 4th ACM Symp. on Computational Geometry*, pages 319–328, 1988.

[ST86] N. Sarnak and R. E. Tarjan. Planar point location using persistent search trees. *CACM*, 29(7):669–679, 1986.

[Sur86] S. Suri. *Computing the Geodesic Diameter of a Simple Polygon.* Tech. Report JHU/EECS-86/08, Dept. of Elec. Eng. and Comp. Sci., Johns Hopkins University, 1986.

[Sur87] S. Suri. Computing geodesic furthest neighbors in simple polygons. In *Proc. 3rd ACM Symp. on Computational Geometry*, pages 64–75, 1987. To appear in *Journal Computer and Systems Sciences.*

[TDS] C. D. Thomborson, L. L. Deneen, and G. M. Shute. A uniform representation for partially embedded graphs. Manuscript.

[Tou86] G. Toussaint. An optimal algorithm for computing the relative convex hull of a set of points in a polygon. In *Signal Processing III: Theories and Applications, Proc. of EUSIPCO-86, Part 2*, pages 853–856, North-Holland, 1986.

[WS87] C. Wang and L. Schubert. An optimal algorithm for constructing the Delaunay triangulation of a set of line segments. In *Proc. 3rd ACM Symp. on Computational Geometry*, pages 223–232, June 1987.

[WS88] A. Wiernik and M. Sharir. Planar realization of non-linear Davenport-Schinzel sequences by segments. *Discrete Comput. Geom.*, 3:15–47, 1988.

[Yap87] C. K. Yap. A $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete Comput. Geom.*, 2:365–394, 1987.

Index

NYU COMPSCI TR-429
Aronov, Boris
Combinatorial and
algorithmic analysis of
space... c.2